DESIGN AND ANALYSIS OF COMMON CONTROL CHANNELS IN COGNITIVE RADIO AD HOC NETWORKS

A Thesis
Presented to
The Academic Faculty

by

Brandon Fang-Hsuan Lo

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2013

DESIGN AND ANALYSIS OF COMMON CONTROL CHANNELS IN COGNITIVE RADIO AD HOC NETWORKS

Approved by:

Dr. Ian F. Akyildiz, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Gordon L. Stüber
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Geoffrey Ye Li
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Manos M. Tentzeris
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mostafa H. Ammar
School of Computer Science
College of Computing
Georgia Institute of Technology

Date Approved: October 31, 2013

To my parents and my wife,

for their love and support.

ACKNOWLEDGEMENTS

When I started my doctoral studies several years ago, I could never imagine that I would have such a memorable and fascinating journey. As my journey approaches an unforgettable end for embarking on a new chapter in life, I owe so many people a debt of gratitude for helping me achieve this goal.

Most important of all, I would like to thank Professor Akyildiz for introducing me this intriguing research field of studies on cognitive radio networks, and providing me the opportunities to grow as a researcher in addition to his guidance, patience, support as well as his wisdom, visions, and real-life experiences that widely open my scopes, deeply sharpen my senses, and profoundly enrich my knowledge and skills throughout my studies. I would also like to thank Professor Gordon L. Stüber, Professor Geoffrey Ye Li, Professor Manos M. Tentzeris, and Professor Mostafa H. Ammar for kindly serving on my committee and providing invaluable feedback that significantly enhances the quality of this work.

I would also like to express my gratitude to former and current BWN Lab members for their assistance and encouragement during the development of this work. Specifically, I would like to thank Dr. Won-Yeol Lee and Dr. Kaushik Chowdhury for their assistance during the early development of this work. Special thanks to Dr. Yahia Tachwali, Dr. Angela Sara Cacciapuoti, and Dr. Marcello Caleffi for their technical discussions, brainstorming, and collaborations as well as their encouragement and friendship to make this work happen.

Last but not least, I would like to thank my family, especially my parents and my wife. This amazing journey would not even have got started or finished without their confidence in me, boundless love, and unconditional support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

CCC Common Control Channel, p. 2.

CR Cognitive Radio, p. 1.

CSS Cooperative Spectrum Sensing, p. 3.

DoS Denial of Service, p. 4.

ERCC Efficient Recovery Control Channel, p. 5.

FC Fusion Center, p. 7.

FCC Federal Communications Commission, p. 1.

ISM Industrial, Scientific and Medical, p. 1.

JRCC Jamming-Resilient Control Channel, p. 5.

MAC Medium Access Control, p. 11.

MARL Multi-Agent Reinforcement Learning, p. 8.

MU Malicious User, p. 4.

PHY Physical Layer, p. 31.

PU Primary User, p. 1.

RF Radio Frequency, p. 2.

RLCS Reinforcement Learning-Based Cooperative Sensing, p. 5.

ROC Receiver Operating Characteristic, p. 101.

SINR Signal-to-Interference-Plus-Noise Ratio, p. 111.

SNR Signal-to-Noise Ratio, p. 77.

SU Secondary User, p. 1.

TVBD Television (TV) Band Device, p. 9.

TVWS Television (TV) White Space, p. 8.

UWB Ultra Wideband, p. 13.

SUMMARY

Common control channels in cognitive radio (CR) ad hoc networks are spectrum resources temporarily allocated and commonly available to CR users for control message exchange. With no presumably available network infrastructure, CR users rely on cooperation to perform spectrum management functions. On the one hand, CR users need to cooperate to establish the common control channels, but on the other hand, they need to have common control channels to facilitate such cooperation. This chicken-and-egg problem, known as the control channel problem, is further complicated by the impacts of primary user activities, channel impairments, and intelligent attackers. Therefore, how to reliably and securely establish control links in cognitive radio ad hoc networks is a challenging problem. In this work, a framework for common control channel design and analysis is proposed to address control channel reliability and security challenges for seamless communication and spectral efficiency in CR ad hoc networks. Specifically, the framework tackles the problem from three perspectives: (i) responsiveness to primary user activities: an efficient recovery control channel (ERCC) method is devised to efficiently establish control links and extend control channel coverage upon primary user's return while mitigating the interference with primary users, (ii) robustness to channel impairments: a reinforcement learning-based cooperative sensing (RLCS) method is introduced to improve cooperative gain and mitigate cooperation overhead such as the effect of control channel fading, and (iii) resilience to jamming attacks: a jamming-resilient control channel (JRCC) method is developed to combat jamming under the impacts of primary user activities and spectrum sensing errors by leveraging cooperative intrusion defense strategies. This research is particularly attractive to emergency relief, public safety, military, and commercial applications where self-organizing CR users are highly likely to operate in spectrum-scarce or hostile environment.

CHAPTER I

INTRODUCTION

1.1 Control Channel Challenges

People today enjoy using broadband wireless devices anytime anywhere for business, entertainment, and social networking, to name a few, in unprecedented ways. The exponential growth of such a strong demand for seamless and reliable wireless services requires more spectrum for new wireless broadband services in the future. Nevertheless, like other natural resources, spectrum is limited in nature. One the one hand, the unlicensed industrial, scientific and medical (ISM) bands are crowded with mutual-interfering wireless devices and services. On the other hand, as reported by FCC Spectrum Policy Task Force [28], unoccupied or frequently idle licensed spectrum, known as spectrum holes or white spaces, can be observed at various locations and time periods. These white spaces are unavailable for spectrum-demanding unlicensed services due to the fixed spectrum assignment policy enforced by the government, which results in inefficient utilization of licensed spectrum. Thus, how to efficiently utilize the spectrum resources becomes an important and imminent issue that motivates the research on cognitive radio (CR) networks [1, 2, 37].

As an enabling technology and the promising solution to resolve the spectrum utilization issue, CR networks enable unlicensed users to access idle licensed bands by opportunistic spectrum access [102]. To realize this, unlicensed users equipped with CRs, known as CR users or secondary users (SUs), are capable of detecting the presence of licensed users, also known as primary users (PUs), in licensed bands, and utilizing those spectrum opportunities for their transmission when PUs are not present. To detect PUs or spectrum holes, CR users observe primary transmission,

known as radio-frequency (RF) stimuli from the radio environment, by *spectrum sensing*. Upon the detection of PUs' presence, CR users either adapt themselves to limit their interference with PUs to a tolerable level, or vacate the channel to protect PUs from harmful interference. In the latter, CR users need to determine an appropriate frequency band for transmission based on spectrum characteristics by *spectrum decision* and resume their transmission in a new band by *spectrum mobility*. In either case, CR users utilize the spectrum efficiently to improve system performance by *spectrum sharing*. These four spectrum management functions form a *cognitive cycle* [1]. To facilitate these spectrum management functions, CR users usually coordinate with each other by using a common medium for control message exchange. This common medium is known as a common control channel (CCC) [1, 2, 57].

A CCC in CR networks is an indispensable medium allocated in a portion of spectrum commonly available to two or more CR users for control message exchange. The CCC allocation can be temporary or permanent in a licensed or unlicensed band to facilitate various CR network operations such as transmitter-receiver handshake, neighbor discovery, channel access negotiation, topology change and routing information updates, and cooperation among CR users [1, 2, 57]. Specifically, CR users show their existence by broadcasting control messages on the CCC for neighboring users in the proximity to maintain the contact and network connectivity [57]. Moreover, CR users can cooperate and share their spectrum sensing data with each other by using the CCC to improve the detection of PUs [3]. More importantly, CR users need to inform each other about PU activity changes, spectrum availability, and network topology in order to improve the CR throughput and spectrum efficiency. However, despite their ubiquitous use, the existence of reliable CCCs are assumed to be constantly available in a significant amount of CR solutions in the literature [5]. In many existing solutions, however, the issues of how CCCs are reliably established and efficiently maintained in the dynamic environment affected by PU activity are often

ignored. Thus, it is essential to investigate the CCC reliability issues and provide novel CCC solutions to address these new challenges in CR networks.

The CCC design in CR networks faces several new challenges. These challenges arise from unique characteristics in CR networks such as PU activity, spectrum heterogeneity, and intelligence of CR users. First, unless allocated in the frequency band free from PUs, a CCC is susceptible to PU activity and can be occupied by PUs at any given time. In this case, the control channel problem in CR ad hoc networks is referred to as a chicken-and-egg problem [5]: CR users need to cooperate with each other to find a PU-free CCC to avoid the interference with PUs. However, they also need to have a PU-free CCC in the first place to facilitate their cooperation and control message exchange. Thus, upon PU's return, CR users face the difficulty in establishing a new CCC without a CCC because they are unable to use the original CCC to negotiate a new one. How to efficiently respond to PU activity and recover the CCC becomes the most important challenge in CCC design. Second, unlike legacy multi-channel wireless networks where all channels are at the disposal of all users, CR users usually observe different channel availability that only a subset of all licensed channels are available. Due to this spectrum heterogeneity in CR networks, it is unlikely to find a channel commonly available to all users as the CCC. As a result, the area where CR users share the same CCC, called *CCC coverage*, is limited to a neighborhood in CR ad hoc networks. Since broadcasting on CCCs of small coverage in the network increases channel switching delay and control signaling overheads, another design challenge is to improve CCC coverage for control message broadcasts under spectrum heterogeneity.

In addition to PU activity and CCC coverage challenges, unreliable control channel conditions can have the great impact on the performance of CR networks. This is a critical issue for cooperative spectrum sensing (CSS) [3] in CR networks. In cooperative sensing, CR users rely on a reliable CCC to report local spectrum sensing

data to a data fusion center or share the data among themselves. However, channel impairments such as multipath fading and shadowing in control channels cause errors in the reported sensing data, which can significantly compromise the detection performance. Moreover, unreliable control channels can result in long reporting delay due to packet loss and retransmission. The increased sensing time due to these delays results in reduced transmission time, which degrades the system performance. Thus, channel impairment is a critical issue of CCC reliability in CR networks. Furthermore, control channels, considered as a single point of failure when statically allocated, are susceptible to security attacks such as control channel jamming. Jamming attacks are launched by malicious users (MUs) to deliberately disrupt the communications of CR users, resulting in denial of service (DoS). It is reported in [15] that control channel jamming can be more effective than jamming the entire band by several orders of magnitude. For this reason, attackers may prefer control channel jamming than other jamming methods due to its effectiveness of resulting in DoS. The jamming issues in CR networks are further complicated by the intelligence of the attackers. Equipped with CRs, these malicious attackers are capable of learning channel allocation strategies of normal CR users and adapting to the behavior of CR users for effective jamming. Thus, as in any wireless network, control channel jamming is a severe CCC reliability issue in CR networks.

In this research, we focus on the CCC reliability issues in CR ad hoc networks. CR ad hoc networks [1] are distributed CR networks formed by a group of CR users connected in an ad hoc fashion without network infrastructure and centralized control entity such as base stations (BS). The reasons of tackling CCC issues in CR ad hoc networks are threefold: (i) CCCs are so crucial in CR ad hoc networks because CR users totally rely on CCCs to cooperate with each other in order to perform all spectrum management functions, (ii) the CCC issues in distributed networks are

more challenging than those in centralized ones simply due to the lack of a centralized control entity for coordination, and (iii) many important commercial, military, emergency relief, and strategic situation applications, which either do not have network infrastructure available or prefer self-organizing networks, strongly demand distributed CCC solutions. Therefore, motivated by the aforementioned CCC issues and challenges in CR ad hoc networks, we present the research objectives and solutions of the research in the next section.

1.2 Research Objectives and Solutions

The objectives of this research are to address three main CCC reliability issues: (i) responsiveness to PU activities, (ii) robustness to channel impairments, and (iii) resilience to jamming attacks. A framework for CCC design and analysis is constructed to address these CCC challenges for seamless communication and spectral efficiency in CR ad hoc networks. As shown in Figure 1.1, the framework consists of three CCC design and analysis methods, each of which aims at tackling a specific CCC reliability issue: (i) Efficient Recovery Control Channel (ERCC) method [54] for responsiveness to PU activities, (ii) Reinforcement Learning-based Cooperative Sensing (RLCS) method [53, 58] for robustness to channel impairments, and (iii) Jamming-Resilient Control Channel (JRCC) method [55, 56] for resilience to jamming attacks. These control channel solutions are discussed as follows.

1.2.1 Responsiveness to Primary User Activities

The first objective of this research is to address the issue of responsiveness to PU activities. This objective aims to efficiently recover CCCs among a large group of CR users upon the return of PUs. This will facilitate virtually "always-on" CCCs in the highly dynamic RF environment to ensure network connectivity and seamless operations, which is especially important for CR ad hoc networks. While the CCC coverage is increased by rendezvous of a large group of CR users, the interference with

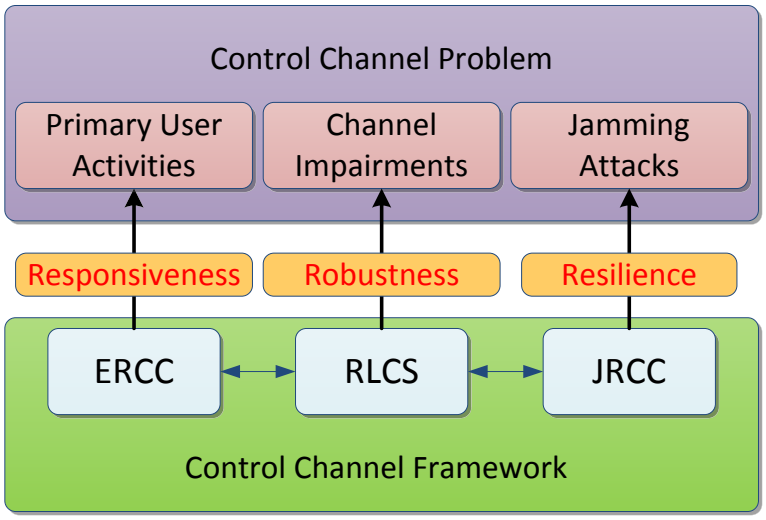


Figure 1.1: Framework of Common Control Channel Design and Analysis.

PUs may be deteriorated due to spectrum heterogeneity. Therefore, the key challenge to achieve the responsiveness of CCCs lies in the tradeoff between maximizing the CCC coverage and minimizing the interference with PUs.

To achieve this objective, we devise an efficient recovery control channel (ERCC) method to efficiently recover CCCs by dynamic control channel allocations while maximizing the CCC coverage for reduced control signaling efforts. Specifically, ERCC enables CR users to prioritize available channels based on local spectrum sensing data and neighbors' preference for finding the best CCC candidate that is the most preferable by the majority of CR users in the neighborhood in preparation for instant recovery of CCCs upon PU's return. As a result, this method effectively recovers lost control channel links caused by PU activity changes and maintains a high degree of network connectivity. Furthermore, ERCC is capable of extending the coverage of a CCC while allocating a control channel of high quality to minimize the interference with PUs. Therefore, ERCC balances the tradeoff between coverage and interference, which facilitates broadcasts with reduced control signaling efforts and increased broadcast throughput.

1.2.2 Robustness to Channel Impairments

The second objective of this research is to provide robustness to control channel impairments in the context of cooperative spectrum sensing. Cooperative sensing is an effective method to combat multipath and shadow fading, and improve spectrum sensing performance by exploiting the spatial diversity of spatially distributed CR users [3]. However, spatially correlated shadowing in sensing channels or reporting channels can limit the achievable cooperative gain [26, 32]. Moreover, the reporting delays incurred by uncorrectable errors in control packets and retransmission need to be minimized to reduce the total sensing time and increase CR throughput. Thus, a new method to select uncorrelated CR users for cooperation and minimize the impact of cooperation overhead is desired.

To achieve this objective, we introduce a reinforcement learning-based cooperative sensing (RLCS) method to provide robustness to channel impairments and improve the detection performance under correlated shadowing and control channel fading. In RLCS, the CR user acting as the fusion center (FC) is the decision-making agent interacting with the environment that consists of its cooperating neighbors and their observations of PU activity. By using proposed reinforcement learning algorithms, the FC learns the behavior of cooperating SUs and takes action to select the optimal set of spatially uncorrelated users for cooperation with the minimum reporting delays. In addition, RLCS is able to adapt to environmental change such as CR user movements, PU activity changes, and varying channel conditions, and mitigate their impact on the performance of cooperative sensing.

1.2.3 Resilience to Jamming Attacks

The third objective of this research is to provide resilience to control channel jamming attacks and maintain network connectivity in hostile environment. As previously

mentioned, control channel jamming is a DoS attack that can effectively disrupt normal network operations. In CR networks, the attackers are also intelligent decision makers who can observe control channel allocations of CR users and select optimal jamming strategies to maximize the effects of jamming while minimizing their consumed energy. Since the establishment of control channels relies on the cooperation of CR users and the availability of control channels for cooperation diminishes under jamming, a new method to combat control channel jamming attacks while sustaining network connectivity for cooperation is necessary.

To address this problem, we develop a jamming-resilient control channel (JRCC) method to provide resilience to jamming attacks launched by intelligent attackers in hostile environment. By using enhanced multiagent reinforcement learning (MARL) algorithms with variable learning rates, CR users can make independent decisions to facilitate future control channel allocations as well as mitigate the effects of jamming attacks to maintain network connectivity. In addition, JRCC is able to adapt to PU activity and mitigate the impact of spectrum sensing errors by exploiting CR user cooperation such as action strategy coordination, best-effort cooperative sensing, and scalable CR user deployment as intrusion defense strategies, which is shown to significantly improve jamming resilience of CR users and compromise jamming strength of attackers.

1.3 Applications of Common Control Channel Solutions

Our CCC solutions are particularly attractive to emergency relief, military, and commercial applications where CR users are self-organized and connected in an ad hoc fashion with no presumably available network infrastructure, and are highly likely to operate in spectrum-scarce or hostile environment. As an example, our CCC design framework can be utilized to manage the coexistence of heterogeneous networks and enhance spectral efficiency in TV white space (TVWS).

TV white spaces are unutilized spectrum resources or frequencies not operated by the licensed devices in the TV bands [30]. To improve spectrum utilization, Federal Communications Commission (FCC) adopts rules to allow unlicensed access in the TV bands [29]. Many standardization activities either finalize new TVWS standards such as IEEE 802.22 [38] and ECMA 392 [27] to enable new TV band devices (TVBD), or extend existing wireless standards such as IEEE 802.11 and IEEE 802.15.4 to enable existing wireless devices for TV band access. However, these emerging TV band devices from heterogeneous networks can result in severe interference with each other due to propagation characteristics of transmissions in the TV bands and the lack of inter-network communication or cross-network coordination. Therefore, the coexistence of heterogeneous networks is envisaged to be a challenging issue in TVWS, and common control channel access has been identified as an open problem and potential solution for TVWS standards [33].

To address the heterogeneous coexistence problem, we consider a CR ad hoc network formed by base stations (BS), access points (APs), personal area network (PAN) coordinators, known as fixed or Mode II devices, and the associated personal/mobile devices in TVWS. These TVBDs (CR users) are able to find neighbors and communicate with each other by using self-organizing ERCCs dynamically allocated in TV channels. For coexistence, each CR user performs RLCS to detect the presence of other TVBDs and estimate channel conditions and interference in the neighborhood. By communicating on the established CCCs, these TVBDs of different networks are able to directly share the environment information such as spectrum sensing data, channel allocation, and transmission power from neighboring heterogeneous CR users and from the TVWS database. As a result, these CR users can iteratively keep track of the environmental change and adaptively adjust their channel and power allocation to mitigate the interference in TVWS. When these TVBDs are at risk of jamming attacks, JRCC can be utilized to effectively establish CCCs and significantly enhance

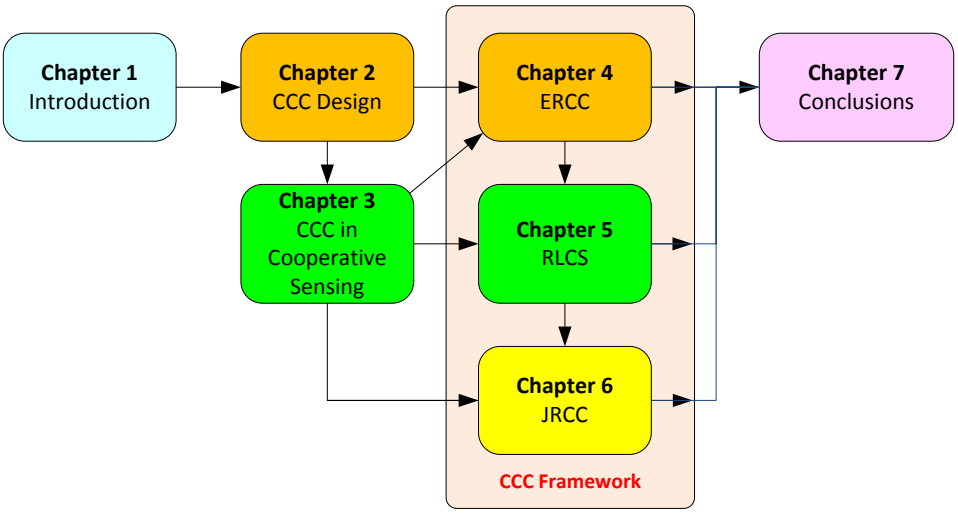


Figure 1.2: Organization of the Thesis.

jamming resilience in TVWS.

1.4 Thesis Outline

In Figure 1.2, the thesis outline and the suggested reading sequences are illustrated. The remainder of this thesis is organized as follows. In Chapter 2, the classification of CCC design methods, major CCC design methods, and their design challenges are discussed. In Chapter 3, cooperative spectrum sensing (CSS) and the design challenges of CCC as reporting channels in cooperative spectrum sensing are introduced. In Chapter 4, the ERCC method is devised for responsiveness to PU activities and the balance of increasing CCC coverage and reducing interference. In Chapter 5, the RLCS method is introduced for robustness to channel impairments and efficient use of CCC bandwidth in cooperative spectrum sensing. In Chapter 6, the JRCC method is presented for resilience to jamming attacks, and the large-scale jamming-resilient control channels under the effects of primary user activity and sensing errors are analyzed. In Chapter 7, the conclusions and contributions of this work and directions for future research are summarized.

CHAPTER II

COMMON CONTROL CHANNEL DESIGN

2.1 Origins of Common Control Channel Design

The CCC design in CR networks is originated from the medium access control (MAC) protocols in multi-channel wireless networks. In multi-channel environment, one channel available to all nodes in the network is commonly used for control message exchange to facilitate negotiations for channel access, handshaking between transmitters and receivers, and other network operations. However, such a single and dedicated control channel allocation may suffer from *control channel saturation* [81] when a large number of nodes access the control channel and cause throughput degradation due to control packet collisions. To address this problem, more flexible control channel allocation schemes were proposed for MAC protocols in multi-channel wireless networks [64]. These early control channel studies for MAC protocols in legacy wireless networks pave the way for the CCC design in CR networks.

In the early studies of MAC protocols for CR networks, control channel solutions remain part of MAC protocols. In fact, a significant amount of CR MAC protocols [36, 39, 48, 65, 83] continue to assume that a dedicated control channel free from PU activity is available to all CR users. This assumption, though simplifying these MAC problems, requires the allocation of control channels either in bands licensed by CR network operators or in unlicensed bands. On the one hand, the control channel, when allocated in licensed bands, incurs undesirable operating cost on the CR network operators. On the other hand, control channels, when allocated in unlicensed bands, can be unreliable due to the interference with other devices of any wireless networks operating in the overcrowded unlicensed bands. More importantly, such

static control channel allocations can result in inefficient spectrum utilization, which contradicts the objective of improving spectral efficiency in CR networks. Thus, it is necessary to devise novel CCC solutions to address new design challenges in CR networks.

In this chapter, we introduce the definition of a CCC and the classification of CCC design methods in CR networks in Section 2.2. Based on the classification, we then discuss major CCC design methods, their pros and cons, and exemplary solutions in Section 2.3. Lastly, we discuss CCC design issues from control channel security perspectives in Section 2.4. Interested readers may refer to [57] for comprehensive surveys of CCC design methods and the extensive discussions of their advantages, disadvantages, and design challenges.

2.2 Definition and Classification

The unique characteristics of spectrum heterogeneity and challenges of resource management in CR networks call for a new definition of CCCs different from the conventional view of control channels. A CCC in CR networks is a medium allocated in a portion of spectrum commonly available to two or more CR users for control message exchange. Based on this definition, a CCC can be allocated in a licensed or unlicensed band, and the allocation can be temporary or permanent. Mathematically, we have the following:

Definition 2.1 (Common Control Channel)**.** *A common control channel (CCC)* $c_v \in \mathcal{C}$ *is a channel allocated in a portion of spectrum* $[f_1, f_2]$ *with channel bandwidth* $B_c = f_2 - f_1$ *during the time period* $[t_1, t_2]$ *for control message exchange, where* $\mathcal{C}$ *is a set of available channels for allocations,* f_1 *and* f_2 *are RF frequencies that satisfy* $3\,\text{kHz} \leq f_1 < f_2 \leq 300\,\text{GHz}$*, and* t_1 *and* t_2 *are time instants that satisfy* $0 < t_1 < t_2 < \infty$*.*

Based on this definition, a CCC in CR networks may not be always available or unique, and it can be allocated in either licensed or unlicensed frequency bands. In

this research, we focus on common control channels dynamically allocated in licensed spectrum where PUs are likely to occupy the allocated control channels anytime, which makes it more challenging to tackle the control channel problem. Several existing CR MAC solutions [44, 45, 98] claim that no CCC is required or needed in their schemes. However, what is not required in those solutions, according to our definition, is a statically allocated CCC, or to be more appropriately termed, a *dedicated CCC*. Thus, by our definition, at least one CCC is always utilized by any MAC or channel allocation schemes in CR networks.

The classification of CCC design is the best place to understand the CCC design in CR networks from the bird's-eye view. The CCC design schemes have been classified in several ways in the literature [1, 57, 69, 70]. As shown in Fig. 2.1, the CCC design classification is first divided into overlay and underlay CCC schemes. This first-level categorization reflects two primary spectrum sharing approaches in the CR paradigm. Overlay approaches are classified as in-band and out-of-band schemes as in [1]. In terms of CCC coverage, in-band approaches are local while the out-of-band schemes are mainly global. The in-band schemes are further classified as sequence-based and group-based CCC designs. The out-of-band schemes are primarily dedicated CCC solutions. Underlay approaches, on the other hand, are composed of ultra wideband (UWB) and multicarrier spread spectrum (MC-SS) underlay control channel designs.

2.2.1 Overlay vs. Underlay

Overlay and underlay approaches are distinguished by how CR users share the spectrum with PUs. In overlay approaches, CCCs are allocated to the spectrum not used by PUs. When the allocated CCCs are occupied by returning PUs, CR users must vacate the CCCs and reestablish new CCCs in other available spectrum. However, the performance of overlay approaches is determined by how fast and accurate the

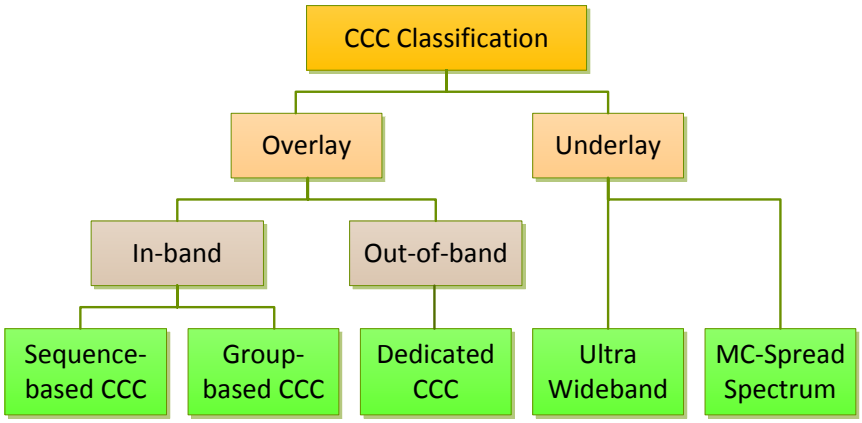


Figure 2.1: Classification of Common Control Channel Design Methods.

PUs are detected and the agility of CCC migration upon PU's return. In underlay approaches, CCCs can be allocated to the same band used by PUs. By utilizing spread spectrum techniques, control messages are transmitted in low power by using short pulses, which are spread over a large bandwidth such that control transmissions appear to PUs as noise. However, PU transmissions can still be affected by underlay CCCs if the number of CR users is large due to the increase of the noise floor.

2.2.2 In-Band vs. Out-of-Band

In-band and out-of-band approaches are determined by whether or not the spectrum for CCC allocations is used by PUs. As a result, the CCCs allocated to licensed channels used by PU transmissions are called in-band CCCs while the CCCs allocated in dedicated spectrum such as unlicensed bands or the spectrum licensed to CR network operators are called out-of-band CCCs. In-band CCCs generally improve spectrum efficiency and control channel security at the cost of establishment overhead and higher complexity. Moreover, the coverage of in-band CCCs is limited to local areas due to the spectrum heterogeneity caused by PU activities. On the contrary, out-of-band CCCs are always available and can be globally available, but they incur extra cost for CR network operators if allocated in licensed bands, suffer from interference if allocated in unlicensed bands, and are susceptible to security attacks.

2.3 Control Channel Design Methods

Based on the CCC classification, we now discuss the following four major control channel design methods: *sequence-based* [6, 8, 9, 22, 23, 88], *group-based* [18, 19, 46, 52, 54, 101], *dedicated* [21, 36, 39, 83], and *underlay* [14, 61, 62, 71, 77, 96].

2.3.1 Sequence-Based Control Channel Design

In sequence-based CCC approaches, control channels are allocated according to a random or predetermined channel hopping sequence. The primary goal of this design is to diversify the control channel allocation over time and frequency spaces in order to minimize the impact of PU activities. Since CR users may use different hopping sequences, different neighboring pairs in a neighborhood communicate on different control channels. As a result, this approach reduces the number of control channels affected by PU's return at a given time. However, such link-based rendezvous mostly between a pair of CR users does not provide large CCC coverage. Thus, sequence-based approaches incur high signaling overhead during control message broadcasts.

Figure 2.2 illustrates sequence-based CCCs in spatial and temporal-frequency domains. Figure 2.2(a) is a spatial-domain snapshot taken between time instant t_1 and t_2 in Figure 2.2(b) that shows PU activities and CCC allocations over time and frequencies. The control links are established in pairs when two CR users rendezvous on the same channel such as CR user I and J communicating on Channel (Ch) 3. Since the CCC coverage is limited to one neighbor at a time, it takes time to meet all neighbors one by one on different channels for a single broadcast message. The blue arrows indicate channel switches following a possible channel hopping sequence $\{3, 2, 5, 6, 4\}$ adopted by CR user I. Note that CR user G and H tuned to Channel 1 cannot communicate with each other due to PU activities on that channel.

In the sequence-based CCC design, the channel hopping sequence is the key element for dynamic channel access. In addition to fixed channel hopping patterns [44],

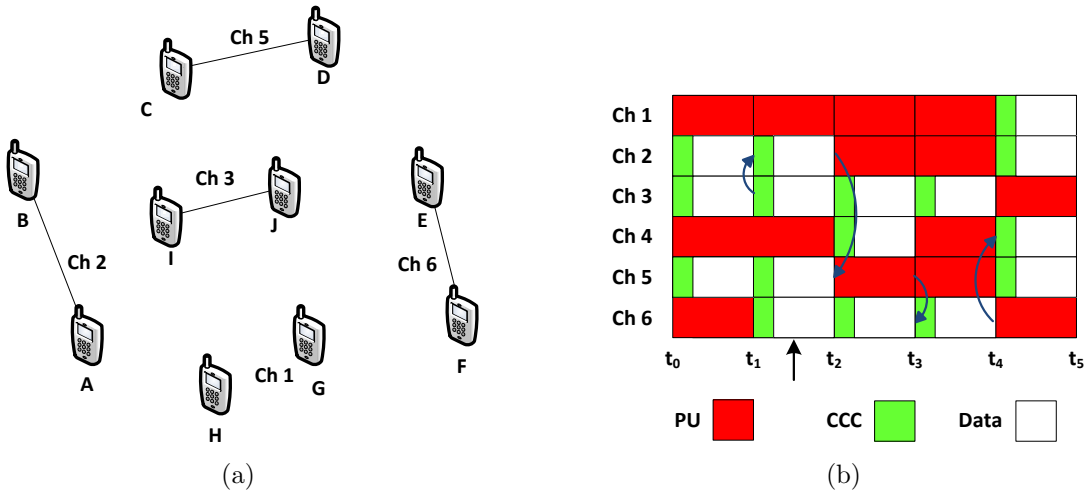


Figure 2.2: Sequence-Based Common Control Channels in (a) Spatial Domain and (b) Temporal-Frequency Domain.

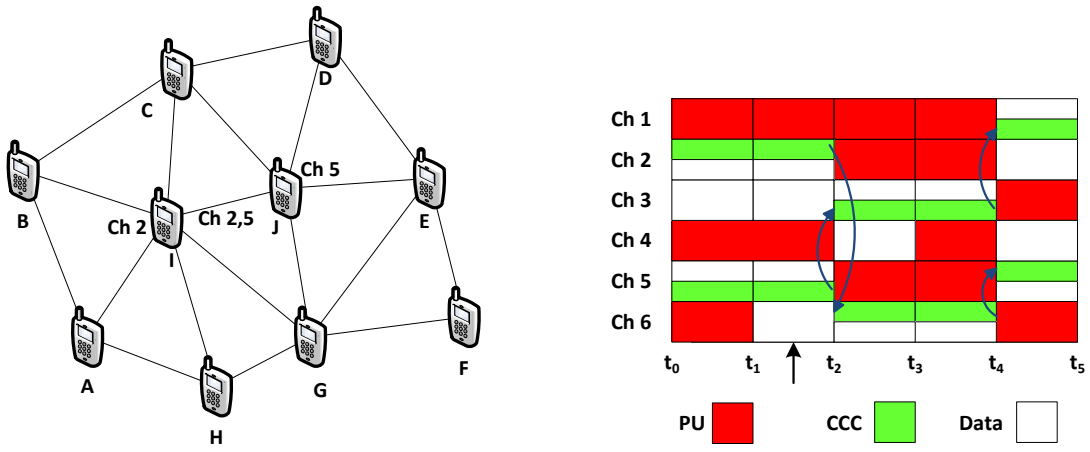
the construction of hopping sequences can be pseudo random [6, 88], permutation-based [23, 88], adaptive frequency hopping [22, 54], or quorum-based [8, 9]. Since the time for two CR users to meet on a channel, known as time to rendezvous (TTR), can be unlimited for random channel hopping, the permutation-based sequence [23, 88] provides the bound on TTR by utilizing certain ordering of the selected channels. The adaptive channel hopping [22, 54] further increases the probability of rendezvous by allocating longer slots to the channels of higher quality. Alternatively, the quorum-based sequence [8, 9] increases the overlapping of multiple sequences to facilitate the rendezvous of two or more CR users with reduced and bounded average TTR by exploiting the nonempty intersection property of quorum systems. Although sequence-based approaches, compared to other approaches, reduce the impact of PU activity on control channels, they are not immune to PU's return. In fact, the TTR is considerably compromised when some channels in the sequence are occupied by PUs because the sequences are constructed with no consideration of PU activities. Thus, the design of channel hopping sequences is essential to the performance of sequence-based control channel approaches.

2.3.2 Group-Based Control Channel Design

In group-based CCC approaches, control channels are the channels commonly available to a group of CR users in proximity. This can be achieved because CR users usually observe similar spectrum availability in a neighborhood. By grouping CR users that use a common channel as the CCC in a local area, group-based CCC designs facilitate control message broadcasts within the group. As a result, the group-based schemes, compared to sequence-based schemes, can generally achieve better CCC coverage. However, how efficient the group responds to PU activities and security attacks depends on the grouping schemes and algorithms. Moreover, inter-group communication between two groups using different control channels can also be a challenge.

Figure 2.3 illustrates group-based CCCs in spatial and temporal-frequency domains. In Figure 2.3(a), two groups are illustrated: one centered at CR user I on Channel 2 and the other centered at CR user J on Channel 5. CR users I and J can reach all their neighbors simultaneously with a single control broadcast message. CR users such as C and G can be tuned to either channel and be part of either group or both groups depending on their hardware and channel selection capabilities. However, the inter-group communications can be a problem if these CR users and their neighbors in the other group are tuned to different channels. In Figure 2.3(b), CCC allocations over time and frequencies of these two groups are illustrated. The blue arrows also indicate CCC changes of the entire group due to PU's return. It is evident that CCC coverage can be increased and the difficulties in inter-group communications can be eliminated if these two groups are merged on a single control channel.

The groups are commonly formed by either neighbor coordination [18, 54, 101] or clustering schemes [20, 46, 52]. In neighbor coordination approaches [18, 54, 101], CR users autonomously vote for the channel commonly available to the largest number



(a) (b)

Figure 2.3: Group-Based Common Control Channels in (a) Spatial Domain and (b) Temporal-Frequency Domain.

of neighbors and exchange the voting information by broadcast. This distributed voting mechanism enables the largest connectivity in the neighborhood via proper CCC selections. In clustering methods, CR users are divided into clusters based on cluster formation and optimization algorithms by using graph theory techniques such as finding the minimal dominating set [20] and finding the maximum edge biclique [46, 52]. These clustering methods aim to increase CCC coverage by selecting CR users who have the largest number of neighbors to share the largest number of commonly available channels as clusterheads in their neighborhoods. The neighbors of these clusterheads are members of the corresponding clusters. Compared to sequence-based approaches, group-based approaches achieve better CCC coverage with larger overhead for regrouping or cluster reformation when control channels are affected by PUs.

2.3.3 Dedicated Control Channel Design

Dedicated CCCs are control channels predetermined in licensed [21, 36, 39, 48, 83] or unlicensed bands [40, 75]. These dedicated approaches are attractive solutions for several reasons: (i) they are usually unaffected by PU activities and considered always

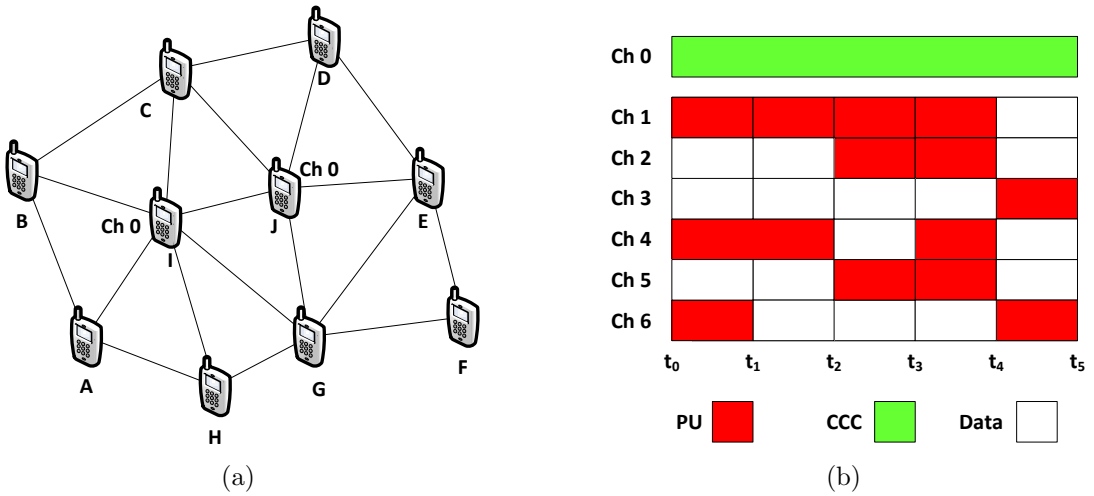
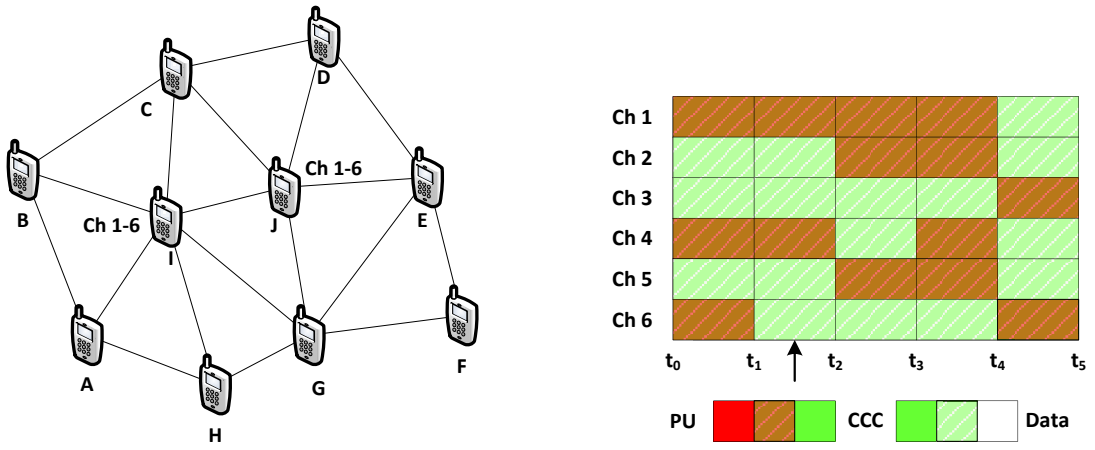


Figure 2.4: Dedicated Common Control Channel in (a) Spatial Domain and (b) Temporal-Frequency Domain.

available ("always on"), (ii) they are available network-wide with global coverage, and (iii) they simplify the design of CR MAC protocols or coexistence protocols. However, dedicated CCCs have disadvantages of possible licensing cost if allocated in licensed bands or severe interference if allocated in unlicensed bands. More importantly, compared to other approaches, dedicated CCCs are more susceptible to control channel saturation [81] and security attacks [57]. Figure 2.4 illustrates dedicated CCCs in spatial and temporal-frequency domains. As shown in Figure 2.4(a) and 2.4(b), Channel 0, not affected by PU activities, is dedicated to control transmission of all CR users with the coverage of the entire network. Nevertheless, it is more susceptible to security attacks due to static allocation and fixed location in the spectrum.

The majority of dedicated CCC solutions in licensed bands are proposed by existing CR MAC protocols such as OSA-MAC [48], Opportunistic MAC [83], and OS-MAC [36]. These CCCs are allocated in a band licensed to CR networks, which are not affected by PU activities at the expense of licensing cost. Alternatively, dedicated CCCs can be allocated by using OFDM subcarriers in the guard bands of the PU licensed spectrum [21], which are only affected by possible adjacent channel interference caused by PU activities. Similarly, dedicated control channels can be allocated



(a) (b)

Figure 2.5: Underlay Common Control Controls in (a) Spatial Domain and (b) Temporal-Frequency Domain.

in unlicensed bands for CR MAC protocols such as HC-MAC [39] and for coexistence protocols such as common spectrum coordination channel (CSCC) [40, 75]. Nevertheless, how to coordinate the access in unlicensed bands to avoid the interference becomes an important issue.

2.3.4 Underlay Control Channel Design

In underlay control channel approaches, spread spectrum techniques such as ultra wideband (UWB) [13, 61, 62, 71, 77] and multicarrier spread spectrum (MC-SS) [96] are utilized to establish control channels occupying large bandwidth with power spectrum that appears to PUs as noise. Figure 2.5 illustrates underlay CCCs in spatial and temporal-frequency domains. As shown in Figure 2.5(a) and 2.5(b), the ultra wideband CCC occupying Channel 1 to Channel 6 is shared by all CR users using different spreading code. The diagonal stripe pattern in the figure illustrates that the underlying CCC appears as noise and does not affect PU activities and data transmission.

In UWB control channel approaches [13, 61, 62, 71, 77], information is modulated on spreading sequences and transmitted in low power as short pulses to exhibit an

ultra wide signal bandwidth compared to channel bandwidth. Since the UWB transmission is perceived as the noise in narrowband channels, this transmission scheme can be utilized to send control traffic in the overlay UWB channel without the harmful interference with the PU traffic in licensed channels. However, the transmission range is limited due to the strict limitation on UWB transmission power. Therefore, UWB CCC design must tackle the following two issues: (i) how to increase the limited transmission range and (ii) how to resolve the range difference between UWB control radios and other types of data radios. In the MC-SS approach [96], the filtered multitone spread spectrum (FMT-SS) technique is utilized for control radio transceiver design. Unlike UWB approaches, FMT-SS control channel design is capable of dynamically masking out subcarriers that correspond to detected PU activities for mitigating the interference with PUs.

2.4 Control Channel Security

While control channels facilitate cooperation among CR users and network operations in CR ad hoc networks, they are exposed to the risks of security attacks. CR users can be vulnerable to a variety of security attacks such as control channel jamming attacks, PU emulation attacks, and data falsification attacks that can interfere with control signal transmission, affect control channel allocations, or manipulate the contents of control messages. Therefore, it is essential that security issues are taken into consideration in common control channel design. We focus on control channel jamming attacks, PU emulation attacks, and integrity of control messages next, and data falsification attacks will be discussed in Chapter 3.4.1.

2.4.1 Control Channel Jamming Attacks

In control channel jamming attacks, strong interference signals are intentionally transmitted by attacks in control channels to interfere the reception and decoding of control messages. Without receiving these control messages, CR users are unable to exchange

control messages in control channels to maintain normal network operations in CR networks such as cooperative spectrum sensing, channel negotiation, and routing information. As a result, control channel jamming is one of the most effective ways to disrupt network operations. Compared to other jamming methods, it is more energy efficient and effective by several orders of magnitude for attackers to cause DoS by control channel jamming [15, 86]. Therefore, designing a control channel scheme resilient to such a DoS attack is crucial to CCC reliability.

Traditionally, spread spectrum techniques are utilized to mitigate jamming attacks by introducing the pseudo random channel access unknown to attackers. However, they become ineffective if any compromised CR user reveals the pseudo-random number (PN) sequences. Moreover, the compromised users cannot be easily identified under jamming. To deal with these problems, there are two main jamming mitigation approaches to combating control channel jamming: (i) dynamic CCC allocation [47, 60] and (ii) CCC key distribution [15, 86, 87]. Although these schemes may not be specifically proposed for CR networks, they can be utilized to mitigate the control channel jamming problem in CR networks.

2.4.1.1 Jamming Mitigation by Dynamic Spectrum Allocation

The dynamic CCC allocation methods combat control channel jamming by dynamically allocating the CCC to maintain the control communication in response to jamming attacks. The dynamic allocation can be achieved by (i) cross-channel communication [60] and (ii) frequency hopping [47].

The cross-channel communication scheme proposed in [60] utilizes the fact that successful communication under jamming attack only requires CR users receiving messages on a channel not affected by the jamming signals. In other words, CR users can continue to transmit on the jammed channel under interference and notify others the new CCC for receiving control messages if the receiving nodes are free of jamming.

As a result, the channels for transmitting and receiving control messages can be different to maintain the control message exchange with neighbors under jamming. Although this scheme provides a mechanism to maintain control communications under jamming, it incurs high channel switching overhead with a single transceiver. In addition, any CR user compromised by the jammer will receive the notification of CCC change and be able to jam the new CCC.

In addition to cross-channel communication, a dynamic control channel allocation scheme based on hopping sequences is proposed in [47] to mitigate the control channel jamming attacks in cluster-based ad hoc networks. In this method, the clusterhead (CH) of each cluster determines the hopping sequences and the operating control channels within the cluster. The affected area is reduced due to the clustering of the network. Since the CCCs are inserted in the sequences, CR users hopping on different sequences in the cluster can rendezvous on the predetermined CCC in the designated time slots without knowing the hopping sequences of others. In addition, the compromised cluster members can be identified if they follow their unique hopping sequences. The hopping sequences are also encrypted by the public key of each CR user to provide the protection from the intruders. However, when the hopping sequences are known to malicious users or compromised users through node capture attack, this method may be ineffective. In this case, all hopping sequences will be known to the jammer and all CCCs will be jammed if the CH is compromised. It can only be resolved by rotating CHs so that new sequences including the designated CCCs are assigned by the new CH. Thus, this method temporarily and intermittently restores the CCC over time and frequency until the jammer is removed.

2.4.1.2 Jamming Mitigation by Control Channel Key Distribution

The second jamming mitigation approach hides CCC locations from the attackers by using the key distribution techniques. In this approach, each authorized user

with a valid key will be able to locate the allocated CCCs by using keyed hash functions. Since the control messages are repeatedly transmitted on multiple CCCs, any compromised nodes having only partial keys in the key space will not be able to jam all the CCCs. Thus, control information exchange can be maintained by sufficiently large key distribution and duplicate messages under jamming attacks.

The jamming-resilient key assignment can be polynomial-based [15] or randomly distributed [86, 87]. In [15], the polynomial-based scheme utilizes the key space consisting of $p \times q$ keys and repeated control transmission by simultaneously sending the control message over q CCCs in each of p time slots in a period. Each user including the malicious ones can be identified by a unique polynomial over Galois field $GF(q)$ with degree $\leq c$. This scheme guarantees at least one CCC access in a period less than $T \log_T N$ time slots with at most $(T \log_T N)^2$ duplicate control messages when T out of N users are compromised and become traitors to jam the CCCs. Since this scheme utilizes the key space size in terms of sufficiently large number of time slots (p) and number of CCCs (q) to combat the jamming by T compromised users, it may incur large control retransmission overhead and delay when T is large. More importantly, the number of traitors T is unknown in advance. As a result, once the number of traitors is greater than a threshold guaranteed by the key space size, the system performance degrades considerably.

To overcome the shortcomings of the polynomial-based scheme, a random key distribution scheme is proposed in [86,87] for CCC access under node capture jamming attacks. Similar to [15], this scheme utilizes the CCC keys to mask the CCC allocation in time slots with duplicate control transmission on multiple CCCs. The random CCC key assignment reduces the risk of the key assignment structure being learned by the attackers. That is, by increasing the diversity of keys assigned to users, authorized users also increase the probability of holding keys unknown to compromised users. However, this method also increases the communication and storage overhead due to

the increase of the number of keys. To limit the key space size and the corresponding storage overhead, the keys are periodic reused in time slots. To prevent the attackers from knowing CCC locations by finding the correlation in transmission patterns, the cryptographic hash functions are used to map the CCC keys to the allocated CCC frequency and time slot for CCC relocation in each key reuse period. Furthermore, the compromised users can be identified by using statistical estimation based on the likelihood of users being compromised.

2.4.2 Primary User Emulation Attacks

In PU emulation attack, malicious users transmit signals similar to those of the PUs. Since these malicious users are mistaken as PUs, legitimate CR users will vacate the frequency band and the attackers will have the wrongful privilege to access the spectrum. Although PU emulation attacks reduce spectrum utilization and the number of channels available for control channel allocations, these attacks are not considered as jamming attacks in this work. This is because the attackers behaving like PUs to preoccupy channels can be detected by CR users just like PU detection and control transmission of CR users can reamin intact if control channels can be established elsewhere. To address PU emulation attacks, a transmitter verification scheme based on localization is proposed in [17] to counter the attack. In this method, an RSS-based localization is utilized by collecting the RSS values from cooperating CR users to estimate the PU transmitter location. The PU identity can be verified by comparing the estimates with known PU characteristics.

2.4.3 Integrity of Control Messages

In addition to control channel jamming and PU emulation attacks that affect the allocations and availability of common control channels, another level of control channel security concerns with the authentication of CR users and the integrity of control

data being transmitted on CCCs. For authentication issues, a CCC security framework is proposed in [76] that includes an authentication phase followed by encrypted transactions for channel negotiation between the transmitter-receiver pair to ensure secure communications on CCCs in CR ad hoc networks. Although this security procedure can prevent eavesdropping and unauthorized access to the CCC, it cannot exclude the access of the compromised users and the manipulation of the control data. For example, CR users share their spectrum sensing data on CCCs to improve the probability of detection in cooperative sensing. The compromised users in this case can manipulate spectrum sensing data in encrypted control messages after passing the authentication. As a result, additional security measures are required to detect these malicious users and their manipulation of control information. Therefore, in this research, we are particularly interested in data falsification attacks on spectrum sensing data reported via common control channels in cooperative spectrum sensing, which will be discussed in Section 3.4.1 of the next chapter.

CHAPTER III

CONTROL CHANNELS IN COOPERATIVE SPECTRUM SENSING

3.1 Control Channels and Cooperation

Common control channels and CR user cooperation are inseparable in CR ad hoc networks. As mentioned in Chapter 1, CCCs facilitate CR network operations because they provide the medium for control message exchange between CR users required by network operations in different network protocol layers. In physical (PHY) layer, CCCs are used as reporting channels for CR users to share their local spectrum sensing results in cooperative spectrum sensing. In MAC layer, CCCs are used for neighbor discovery, channel negotiation, and transmitter-receiver handshake. In network layer, CCCs are used for broadcasting route updates and topology changes. These CCC applications in different network protocol layers assist CR users in making intelligent decisions to improve system performance and spectral efficiency in CR ad hoc networks.

In this chapter and Chapter 5, we focus on cooperative spectrum sensing performance and control channel issues in cooperative spectrum sensing. The reasons for this focus are twofold: (i) control channels that play an important role as reporting channels can have significant impact on the performance of cooperative spectrum sensing and (ii) since cooperative spectrum sensing is an effective and indispensable way to improve spectrum sensing performance in CR networks, the performance improvement can benefit the establishment of reliable common control channels. We first introduce cooperative spectrum sensing, its elements, cooperative gain and cooperative overhead in Section 3.2, and then discuss the requirements of control channels

in cooperative sensing in Section 3.3. Lastly, we discuss the security issues in cooperative spectrum sensing with the focus on data falsification attacks on reported spectrum sensing data in Section 3.4. A comprehensive survey of CSS can be found in [3].

3.2 Cooperative Spectrum Sensing

Spectrum sensing is one of the fundamental spectrum management functions in cognitive radio networks. The detection performance of spectrum sensing has significant impact on the system performance of CR networks. However, as shown in Fig. 3.1, many factors such as multipath fading, shadowing, and receiver uncertainty may considerably compromise the detection performance of spectrum sensing performed individually by each CR user. Fortunately, it is unlikely for all spatially distributed CR users to concurrently experience the fading or receiver uncertainty problem. If CR users, most of which observe strong PU signals, can cooperate and share their local sensing results with each other, the combined cooperative decisions derived from the spatially collected observations can overcome the deficiency of poor observations from some CR users. The overall detection performance can be greatly improved by exploiting the spatial diversity of CR users. This is why cooperative spectrum sensing (simply called *cooperative sensing* thereafter) [3, 13, 32, 63] is an attractive and effective approach to combat multipath fading and shadowing, and mitigate the receiver uncertainty problem.

Conventional cooperative sensing is considered as a three-step process: local sensing, reporting, and data fusion. As shown in Figure 3.2, a group of spatially distributed cooperating CR users obtain observations y_i of PU signals by individually sensing the licensed channels. Each cooperating CR user makes local decisions u_i according to binary hypothesis testing (H_1 and H_0 for the presence and absence of PUs, respectively) and forwards them to the fusion center (FC). The FC performs

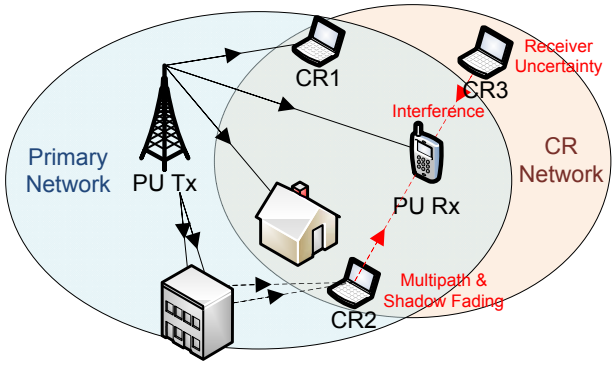


Figure 3.1: Examples of Receiver Uncertainty, Multipath Fading and Shadowing.

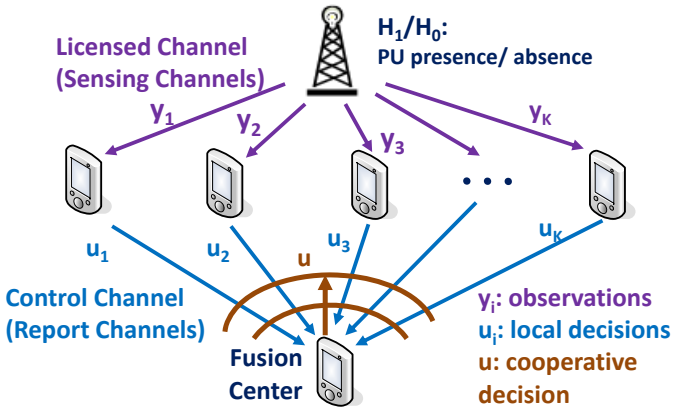


Figure 3.2: Process of Cooperative Spectrum Sensing.

data fusion of reported local sensing data and makes cooperative decisions u. The cooperative decisions, which are generally more accurate than local decisions, are broadcast to all cooperating CR users. From this cooperative sensing process, we can identify the elements of cooperative sensing, which is described next.

3.2.1 Elements of Cooperative Spectrum Sensing

The fundamental components crucial to cooperative sensing process are called the *elements of cooperative sensing*. In our view, the process of cooperative sensing consists of seven key elements: (i) cooperation models, (ii) sensing techniques, (iii) control channel for reporting, (iv) data fusion, (v) hypothesis testing, (vi) user selection, and (vii) knowledge database. These elements are briefly introduced as follows:

- *Cooperation Models* consider the modeling of how CR users cooperate to perform sensing, which includes the popular parallel fusion network models and recently developed game theoretical models.

- *Sensing Techniques* are used to sense the RF environment, take observation samples, and employ signal processing techniques for detecting the PU signal or the available spectrum. The choice of the sensing techniques has the effect on how CR users cooperate with each other.

- *Hypothesis Testing* is a statistical test to determine the presence or absence of PUs. This test can be performed individually by each cooperating user for local decisions or performed by the fusion center for cooperative decision.

- *Control Channel for Reporting* concerns about how the sensing results obtained by cooperating CR users can be efficiently and reliably reported to the fusion center or shared with other CR users via the bandwidth-limited and fading-susceptible control channel.

- *Data Fusion* is the process of combining the reported or shared sensing results for making cooperative decisions. Based on their data type, the sensing results can be combined by using signal combining techniques or decision fusion rules.

- *User Selection* deals with how to optimally select the cooperating CR users and determine the proper cooperation footprint/range to maximize the cooperative gain and minimize the cooperation overhead.

- *Knowledge Database* stores the information and facilitates the cooperative sensing process to improve detection performance. The information in the knowledge base is either a priori knowledge or knowledge accumulated through the experience. The knowledge may include PU and CR user locations, PU activity models, and received signal strength (RSS) profiles.

Based on the elements of cooperative sensing, we can construct the framework of cooperative sensing. The framework consists of the PUs, cooperating CR users including a FC, all the elements of cooperative sensing, the RF environment including licensed channels and control channels, and an optional remote database. Figure 3.3 illustrates the framework of cooperative sensing from the perspective of physical layer (PHY). In this framework, a group of cooperating CR users perform local sensing with an RF frontend and a local processing unit. The RF frontend can be configured for data transmission or spectrum sensing. In addition, the RF frontend includes the down-conversion of RF signals and the sampling at Nyquist rate by an analog-to-digital converter (ADC). The raw sensing data from the RF frontend can be directly sent to the FC or be locally processed for local decisions. To minimize the bandwidth requirement of the control channel, certain local processing is usually required. The processing includes the calculation of test statistics, and a threshold device for local decisions. Once the raw sensing data or local decisions are ready, a MAC scheme is required to access the control channel for reporting the sensing results. The sensing results may also be used by higher network protocol layers. The FC in the framework is a powerful CR user who not only has all the capabilities of a regular CR user, but also the user selection capability with the help of an embedded knowledge database. If the FC is as powerful as a base station, it may have the connection to the remote database for PU activity and white space information. In the case of distributed cooperative sensing in CR ad hoc networks, all CR users individually perform data fusion as a FC. Therefore, they are essentially the same as the powerful CR user in the framework except that the knowledge data base is optional or relatively smaller for local use.

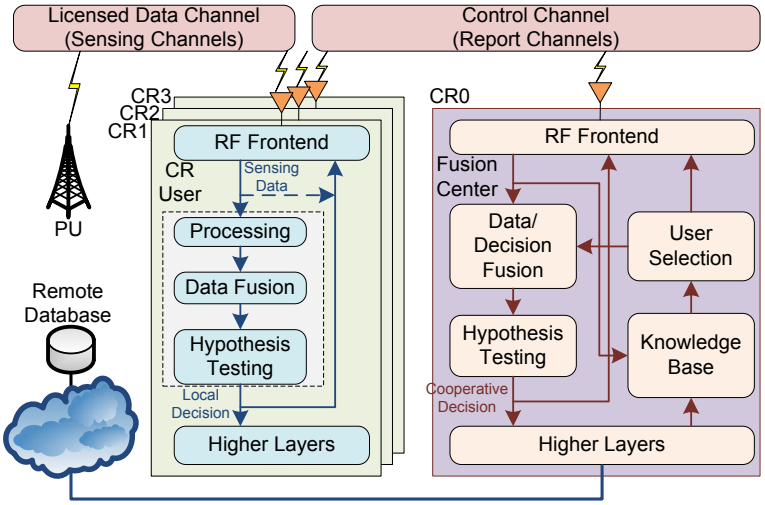


Figure 3.3: Framework of Cooperative Spectrum Sensing.

3.2.2 Cooperative Gain and Cooperation Overhead

The main idea of cooperative sensing is to enhance the sensing performance by exploiting the spatial diversity in the observations of spatially located CR users. The improvement of sensing performance due to spatial diversity is called *cooperative gain*. In addition, cooperative sensing overcomes performance degradation due to multipath fading and shadowing leading to relaxed receiver sensitivity requirements. This is because receiver sensitivity can be approximately set to the same level of nominal path loss without increasing the implementation cost of CR devices [63]. More importantly, CR users can improve their throughput because better sensing performance achieved by cooperative sensing results in less interference and more transmission opportunities. Thus, a well-designed cooperation mechanism for cooperative sensing can significantly contribute to a variety of achievable cooperative gain.

Regardless of the benefits of cooperative sensing, cooperative sensing can incur *cooperation overhead*. The overhead refers to any extra sensing time, delay, energy, security measures, and operation cost incurred by cooperative sensing compared to the individual (non-cooperative) spectrum sensing case. As a result, the achievable cooperative gain can be limited by cooperation overhead. For example, it is known

that more spatially correlated CR users participating in cooperation can be detrimental to the detection performance [32, 63]. Moreover, unreliable or malicious CR users may report falsified data to compromise the sensing performance [16]. Hence, the selection of independent and reliable CR users for cooperation is essentially a form of cooperation overhead. For these reasons, we consider the issues of achievable cooperative gain and incurred cooperation overhead in cooperative sensing as *dominating factors* of the cooperative gain and cooperation overhead, which include (i) sensing time and delay, (ii) channel impairments, (iii) energy efficiency, (iv) cooperation efficiency, (v) mobility, (vi) security, and (vii) wideband sensing. In this research, we focus on tackling CCC-related issues such as sensing time and delay, channel impairments, and security. The challenges are to devise a novel cooperative sensing method effectively leveraged to achieve the optimal cooperative gain without being compromised by the incurred cooperation overhead.

3.3 Control Channel Requirements

In cooperative sensing, control channels, known as reporting channels, are universally utilized to report local sensing data to the FC or share the sensing results with other CR users. In this section, we discuss three major control channel requirements: bandwidth, reliability, and security that need to be satisfied for reporting local sensing data in cooperative sensing.

3.3.1 Bandwidth Requirement

In cooperative sensing, the control channel bandwidth can limit the amount of local sensing data being reported to the FC. Thus, bandwidth requirement determines the level of cooperation [63]. In general, soft combination of raw or quantized local sensing data at the FC achieves better sensing performance than hard combination of binary local decisions at the expense of consuming more control channel bandwidth during reporting. To address the bandwidth issues, many bandwidth efficient

schemes [59, 84, 103] are proposed for cooperative sensing. The bandwidth requirements can be satisfied by censoring the local sensing data for reporting [84], quantizing the local sensing data to fewer bits [59, 84], or superposing local sensing data [103]. The challenges are to achieve satisfactory sensing performance with small required control channel bandwidth.

3.3.2 Reliability Requirement

In addition to bandwidth requirement, the reliability of the control channel has great impact on the cooperative sensing performance. Like data channels, control channels are susceptible to channel impairments such as multipath fading and shadowing. Hence, the effect of control channel impairments must be considered to meet the reliability requirement. Many studies investigate the effects of Gaussian noise [74], multipath fading [100], and correlated shadowing [25, 26] on the control channel and the sensing performance in cooperative sensing. It is shown that the probability of false alarm linearly increases with the probability of reporting errors caused by fading in reporting channels [100]. Moreover, it is found that the performance degradation caused by shadowing correlation in the reporting channel is similar to that in the sensing channel [25, 26]. Therefore, it is important to devise a cooperative sensing scheme to achieve sensing performance as well as mitigating the effect of fading and correlated shadowing.

3.3.3 Security Requirement

The cooperation among CR users raises new concerns for security risks in cooperative sensing. It is reported that the cooperative gain can be severely affected by malfunctioning or malicious CR users in cooperative sensing [63]. First, control channels are subject to jamming attacks. In addition to causing the failure of CR user cooperation, control channel jamming is a form of DoS attacks, which can result in the malfunctioning of the entire network. Second, malfunctioning or malicious CR

users can report unreliable or falsified sensing data to affect or even manipulate the cooperative decisions at the FC. This security risk known as data falsification attack can cause false alarm sensing errors and prevent normal CR users from accessing the available spectrum. To address the security and reliability issues, additional mechanisms such as outlier detection [42, 43], reputation-based mechanism [16, 43], and consensus-based method [99] are required to identify and remove malicious CR users and reported falsified sensing data from cooperation.

3.4 Cooperative Sensing Security

The cooperation among CR users raises new concerns for the reliability and the security in cooperative sensing. This is because, when multiple CR users cooperate in sensing, a few CR users who report unreliable or falsified sensing data can easily influence the cooperative decision. During cooperation, malfunctioning CR users may unintentionally send unreliable data to the FC. For example, the report from a malfunctioning user could deviate from the real value. Moreover, CR users, called malicious users or Byzantine adversaries in this case, can intentionally manipulate the sensing data and report the falsified data for their own benefits. For instance, malicious users may obtain spectrum access by falsely reporting the presence of PUs. It is reported in [63] that cooperative gain can be severely affected by malfunctioning or malicious CR users in cooperative sensing.

To address the security and reliability issues, additional mechanisms are required to identify malicious CR users and manipulated sensing data, and remove them from cooperation. Although these countermeasures may incur overhead in cooperative sensing, they are required to ensure secure operations of cooperative sensing and obtain reliable sensing results in hostile environment. Therefore, we consider two main cooperative sensing security issues: data falsification attacks, where detection performance is affected by the falsified sensing data, and DoS attacks, where cooperative

sensing is disrupted by adversary attacks such as PU emulation and control channel jamming. Since DoS attacks are discussed in Section 2.4, we discuss data falsification attacks next.

3.4.1 Data Falsification Attacks

Data falsification attacks in cooperative sensing, known as spectrum sensing data falsification (SSDF), refer to the attacks of malicious users by reporting falsified spectrum sensing data to FC to manipulate the cooperative sensing decisions for their gain. These are not attacks on common control channel allocations, but on the integrity of control data used in cooperative sensing. To address the data falsification problem, existing cooperative sensing schemes [16,42,43,99] aim to detect the anomaly in the reported sensing data and establish a mechanism to distinguish malicious users from authentic ones so that malicious users can be excluded from the cooperation to ensure the integrity of the sensing decisions.

Specifically, a weighted sequential probability ratio test (SPRT) with reputation-based mechanism is proposed in [16] as the robust cooperative sensing scheme to address the data falsification problem. As the first step, the reputation ratings of cooperating CR users are evaluated based upon their sensing accuracy. The reputation rating is increased whenever the local sensing result matches the final decision, and is decreased otherwise. The reputation values are converted to the weights to be used in the modified likelihood ratio of a SPRT for data fusion. In this manner, the impact of unreliable CR users can be reduced by putting weights on the genuine sensing data over the falsified ones. In addition, a consensus-based cooperative sensing scheme is proposed in [99] to address the data falsification problem in CR ad hoc networks. Each CR user iteratively selects neighbors for cooperation and sensing data exchange such that the consensus (cooperative decision) is gradually reached in a distributed manner. When selecting cooperating neighbors, each authentic CR user checks the

received sensing data by comparing it with the local mean value. The neighbor reporting the result with maximum deviation from the local mean will be rejected for cooperation. With this scheme, the reliability of cooperative sensing can be improved by excluding malicious users from the cooperation in the neighborhood.

Furthermore, a simple outlier detection is proposed in [42] for the pre-filtering of the extreme values in sensing data. The trust factor that measures CR user reliability is then evaluated as the weights in calculating the mean value of received sensing data. In this way, cooperative sensing can be more reliable by building trust toward CR users that report a sensing value close to the mean of all collected results at the FC. The method is extended in [43] to detect malicious users by the outlier factors, which are calculated based on the weighted sample mean and the standard deviation of the energy detector outputs. The outlier factors can be adjusted according to the dynamic PU activities and the observations from the closest neighbors in a neighborhood to further improve the detection of malicious users.

CHAPTER IV

EFFICIENT RECOVERY OF CONTROL CHANNELS

4.1 Motivation

The first challenge of common control channel design in CR ad hoc networks is to address the issue of responsiveness to PU activities. The challenge comes from the dilemma CR users encounter when PUs return to occupy existing CCCs: CR users need to cooperate with each other to find a new PU-free CCC without interfering with PUs, but also need to have a PU-free CCC available to them first to make such cooperation possible. Thus, the focus to resolve this challenge is twofold. First, how CR users can efficiently find their neighbors to establish reliable CCCs and network connectivity when no information about their neighbors and the environment is available in the first place. Second, how CR users can efficiently recover CCCs when CCCs are disrupted by PUs to constantly maintain network topology and connectivity.

Motivated by the design challenges, we introduce efficient recovery control channel (ERCC) method in this chapter to achieve these goals: (i) *Responsiveness to PU Activities:* a new control channel must be immediately established among CR users with no harmful interference with PUs when a PU is present in a control channel, and (ii) *Extended CCC Coverage:* the coverage of a control channel needs to be extended to the largest degree for reducing control signaling overhead and improving broadcast efficiency. ERCC is a heuristic solution that utilizes the spectrum heterogeneity in the environment to improve spectrum efficiency as well as spectrum homogeneity in the neighborhood to facilitate control channel establishment. By prioritizing available channels based on local spectrum sensing data and neighbors' preference, CR users with ERCC are able to find the best CCC candidate that is the most preferable by the

majority of CR users in the neighborhood to establish the connections with neighbors initially and to achieve instant recovery of CCCs among a large group of neighbors in the event of PU's return.

In Section 4.2, we first describe the system model for ERCC design. In Section 4.3, we introduce the proposed ERCC algorithm and show how efficient recovery of CCCs can be achieved by ERCC. In Section 4.4, we establish a theoretical model for performance analysis, analyze CCC recovery and allocation time, and discuss delay, interference, overhead, and other issues. In Section 4.5, we define four CCC metrics for performance evaluation. Finally, in Section 4.6, we evaluate ERCC performance in a variety of test scenarios. The variables and notations used in this Chapter are tabulated in Table A.1 and A.2 of Appendix A for reference. Our contributions are summarized as follows:

- We propose ERCC method that enables efficient recovery of CCCs upon PU's return to CCCs and achieves responsiveness to PU activities. The proposed method effectively recovers lost control links caused by PU activity changes and maintains high degree of network connectivity while achieving the balance of extending CCC coverage and mitigating the interference with PUs.

- We establish the theoretical model for analyzing delay, control throughput, accumulated interference, establishment overhead, and CCC allocation and recovery in which the distributions of CCC recover and allocation time are derived. Moreover, we devise CCC metrics for evaluating ERCC performance in recovery efficiency, CCC coverage, channel quality and PU interference.

4.2 ERCC System Model

For opportunistic spectrum access, a CR ad hoc network is overlaid with a primary network where PUs operate in a set of licensed channels. The number of licensed channels is denoted by N_c. The channels available to each CR user in the CR ad hoc

network may only be a subset of all licensed channels due to PU activities. Thus, CR users rely on local spectrum sensing to observe channel conditions and identify spectrum opportunities.

For spectrum sensing and data transmission, each CR user is equipped with two half-duplex transceivers that can be tuned to any licensed channel. One radio, called control radio, is dedicated to allocated control channels, and the other, called data radio, is used for data transmission. Each radio can transmit data, receive data, or sense a channel, but cannot perform more than one of these operations simultaneously.

The PU or CR user transmit power decays with distance based on the free-space path loss model. When the shadow fading is considered, the combined path loss and shadowing model is used [34]. For correlated shadowing, we use the exponential correlation model [35] and the correlation function is given by $\rho_{ij} = e^{-ad_{ij}}$ [32], where a is the exponential decaying coefficient and d_{ij} is the distance between CR users i and j. To determine the presence of PUs and neighboring CR users within the transmission range, CR users compare sensing thresholds γ_{pu} and γ_{su} with the receive power of PU and CR user signals, respectively. In addition, given the accumulated PU interference level γ_i on channel C_i, $i \in N_c$, the channel quality of C_i is *better* than that of C_j if $\gamma_i < \gamma_j$. As a result, the channel C_i is defined as the *best* channel or the channel of the *best* quality if $i = \arg\min_j \gamma_j, j \in N_c$. For simplicity, other types of fading and interference are not considered in this model.

When a PU returns to a control channel, the control radio ensures the detection of PU signals in a timely manner and switches to a new CCC based on the proposed control channel allocation method. Without the dedicated control radio, a CR user with single radio may be unaware of the control channel change because of using the only radio for data transmission. In addition, the synchronization of quiet periods for spectrum sensing is negotiated among neighboring nodes in the control channel. Thus, energy detection of PU transmit signals can be enforced at the link layer and

performed by the data radio in quiet periods.

PU activities are modeled as a two-state birth-death process [49], an ON-OFF model in which an ON-state represents the appearance of any PU, while an OFF-state represents the absence of all PUs. If the ON state switches to the OFF state with the probability α and the OFF state switches back with the probability β, the steady-state probability of ON and OFF states are $\frac{\beta}{\alpha+\beta}$ and $\frac{\alpha}{\alpha+\beta}$, respectively. Thus, the state transition is a Poisson process while PU interarrivals are exponentially distributed.

In the next section, we introduce our proposed efficient recovery common control channel design.

4.3 Efficient Recovery Control Channel Algorithms

The efficient recovery control channel (ERCC) design consists of three major components: (i) *neighbor discovery*, (ii) *common channel list update*, and (iii) *efficient PU activity recovery*. The neighbor discovery process aims at increasing the probability of locating neighbors on common channels for the establishment of initial network topology. The common channel list update focuses on maintaining a robust list of common channels by using local sensing and neighbor information on a regular basis. The efforts of common channel list updates facilitate the efficient recovery from PU activities in the event of PU's return to the CCC. These components are described in details in the following subsections.

4.3.1 Neighbor Discovery

The system starts with a neighbor discovery process to establish initial network connectivity. During this process, all CR users, initially distributed in a set of predefined licensed channels, locate neighboring nodes within their transmission range. To locate neighbors in the network, each CR user obtains a list of available channels from local observations, follows a channel hopping sequence, and hops over available channels.

A neighboring pair discovers each other and establishes a link when they hop to the same channel and exchange beacon messages. Thus, the initial network topology is formed after all links are established among neighboring nodes. Next, we describe the construction of channel hopping sequence, handshaking procedure, and neighbor list update in the neighbor discovery process.

4.3.1.1 Channel Hopping Sequence

The channel hopping sequence is a pseudo-random sequence of available channels for frequency hopping during neighbor discovery. To construct such a sequence, a CR user starts with a channel list based on local observations of channel availability. The channels in the list are initially of order in decreasing channel quality (known as a preferred channel list in Section 4.3.2). To maximize the chance of locating neighbors in preferred common channels, the control radio is tuned to channels with the preference according to the channel order. For a common channel list L_C of length n (also defined in Section 4.3.2), the probability of selecting $C_i \in L_C$, with bias toward lower index i, is given by:

$$Pr(C_i) = \frac{n+1-i}{\sum_{j=1}^{n} j} = \frac{2(n+1-i)}{n(n+1)}, \quad 1 \leq i \leq n. \tag{4.1}$$

Since $Pr(C_i)$ can be considered as the probability mass function (PMF) of a discrete random variable $C = C_i$, a discrete cumulative distribution function (CDF), denoted by $F_C(C)$, can be obtained accordingly. Thus, the value of the CDF at $C = C_i$ is given by:

$$F_C(C_i) = \sum_{j=1}^{i} Pr(C_j), \quad 0 \leq i \leq n, \tag{4.2}$$

where $F_C(C_0) = 0$ and $F_C(C_n) = 1$. These values in (4.2) are used as thresholds in the mapping from the sequence of random numbers $r_m \in (0, 1]$ to the selected channel C_i. Thus, the channel hopping sequence $S_m, m = 1, 2, \ldots$, is generated as follows:

$$S_m = C_i \text{ for } F_C(C_{i-1}) < r_m \leq F_C(C_i), m = 1, 2, \ldots. \tag{4.3}$$

Since channels with higher preference in the common channel list appear more often in the channel hopping sequence, CR users locate their neighbors in a channel common to more neighbors with higher probability.

4.3.1.2 Handshaking Procedure

In the neighbor discovery process, each CR user follows its hopping pattern and tunes to one channel at a time. During the time interval, the CR user broadcasts a beacon with random backoff and listens to the channel for any beacon broadcast. If the CR user receives a beacon from a neighbor, it replies with an Ack message. Similarly, the CR user receives an Ack if its neighbor receives the beacon. The beacon notifies neighbors of the CR user's ID and its common channel list while the Ack message ensures that the neighbor discovery between a neighboring pair is mutually recognized. Therefore, a link is established in a common channel between the neighboring pair after the beacon and Ack exchanges.

4.3.1.3 Neighbor List Update

After the neighboring pair completes the handshaking procedure, each CR user's neighbor list is updated accordingly. The neighbor k is added to the neighbor list if it is new to the list. The control channel associated with this neighbor, denoted by Ch_k, is updated with the allocated control channel. The allocated CCC may be different from the channel the neighboring pair meets because a channel that can reach more neighbors or has higher quality is preferred.

Since each end of the link obtains its own and neighbor's broadcast common channel lists after beacon exchange, the neighboring pair can individually generate a set of channels, denoted by L_{CC}, from the intersection of those two broadcast common channel lists. The best channel of L_{CC} is allocated as the CCC to the link. Thus, identical decision of this CCC allocation can be individually determined by the neighboring pair based on the same L_{CC}. No further control message exchange is

required.

The neighbor discovery process is highlighted in Algorithm 1. In line 4, t_{disc} is the maximum duration for initial neighbor discovery. Line 7 to 9 outline the handshaking procedure while line 10 to 15 show the neighbor list update and initial control channel assignment. The *OrderChannel* function in line 13 reorders the channels based on the ordering rules, which will be described in next section.

Algorithm 4.1 : Neighbor Discovery

1: Preferred Channel List $L_{Pi} \leftarrow$ LocalSensing(γ_{pu})
2: Common Channel List $L_{Ci} \leftarrow L_{Pi}$
3: $\{S_m\} \leftarrow$ SequenceGenerator(L_{Ci})
4: **while** NbrDiscoverTimer $\leq t_{disc}$ **do**
5: SwitchChannelTo(S_m)
6: RandomBackoff and SendBeacon(i, L_{Ci})
7: **if** ReceiveBeacon(k, L_{Ck}) from neighbor k **then**
8: SendAck(i, L_{Ci}, k)
9: **end if**
10: **if** ReceiveBeaconOrAck(k, L_{C_k}, i) **then**
11: $L_{NBi} \leftarrow L_{NBi} \cup \{k\}$
12: $L_{CC} \leftarrow L_{Ci} \cap L_{Ck}$
13: $L_{Ci} \leftarrow$ *OrderChannel*$(L_{Ci}, L_{CC}, \gamma_j)$
14: $Ch_k \leftarrow \arg\min_j \{C_j | C_j \in L_{CC}\}$
15: **end if**
16: **end while**

4.3.2 Common Channel List Update

Since neighboring CR users usually observe homogeneous channel availability in CR ad hoc networks, each CR user can individually obtain a similar list of available channels. Intuitively, channels in those lists common to a large number of neighboring nodes are the candidates for CCC allocations. Thus, the advantages of maintaining an ordered list of common channels are twofold: (i) selecting the channel common to the largest number of neighbors as the control channel increases the coverage of the CCC and, more importantly, (ii) when a PU occupies the control channel, the common channel with the highest preference from the list can be immediately allocated as

the new control channel. With such an allocation, most neighbors can immediately locate each other in the new control channel. Therefore, efficient recovery from PU activities can be achieved by common channel list updates.

Each CR user constructs and maintains a *common channel list* (CCL) for periodic broadcast to neighbors and dynamic CCC allocations. In general, a CCL is a list of channels commonly available to at least one neighbor. The order of the list is determined by the weight and the quality of the channels. The weight of a channel C_i, denoted by W_i, is the number of neighbors having C_i in their CCL. It indicates the number of neighbors a CR user could reach if the channel is allocated as the CCC. Equivalently, it represents the preference of choosing the channel as a CCC in the neighborhood. Therefore, the channel order of a CCL follows two rules: (i) all channels in the CCL are of monotonically decreasing order according to W_i and (ii) if two channels have the same weight, their order is determined by the PU interference level γ_i. In other words, C_i is preferred to C_j, $i \neq j$, if (i) $W_i > W_j$ or (ii) $W_i = W_j$ and $\gamma_i \leq \gamma_j$.

To construct a list of channels commonly available to neighbors, a CR user requires its own observations of channel availability, a list called *preferred channel list* (PCL), and neighbors' preference of available channels. Therefore, CR users update their CCL when they obtain a new PCL from local sensing or receive a CCL from neighbor's broadcast.

4.3.2.1 CCL Update with Local Sensing Information

To obtain a PCL, a CR user senses each licensed channel, determines PU-occupied ones, and returns with a list of available channels in the order of observed channel quality. A PCL, denoted by L_P, is a channel list of observed quality in monotonically decreasing order. Since the channel quality in channel C_i is inversely proportional to

total receive power of PU transmit signals γ_i, a PCL of length n is defined as:

$$L_P = \{C_i | 1 \leq i \leq n \text{ for } \gamma_1 \leq \ldots \leq \gamma_n \leq \gamma_{pu}\} \tag{4.4}$$

where γ_{pu} is the PU interference threshold that determines PU's presence in a channel. Since PU-occupied channels are excluded from the PCL, all channels in L_P are presumably available unless PUs change their operating location or channel.

After obtaining a PCL from local sensing, a CR user updates its CCL with the PCL. The CCL is initially set to the PCL and successively updated by new PCLs from periodic sensing. The update is essential for the following two reasons: 1) new PU-occupied channels that no longer exist in the PCL should be removed from the CCL. 2) newly available channels should be added to the CCL for neighbor notification. Thus, a CCL after the update reflects the most up-to-date channel conditions.

Mathematically, given a CCL L_C and a PCL L_P, the removal of PU-occupied channels is given by:

$$L_C \leftarrow L_C \setminus \{C_j | C_j \in L_C \text{ and } C_j \notin L_P\}. \tag{4.5}$$

On the other hand, the addition of newly available channels is given by:

$$L_C \leftarrow L_C \cup \{C_j | C_j \in L_P, C_j \notin L_C, \text{ and } W_j = 0\}. \tag{4.6}$$

Notice that the weight W_j associated with the newly added channel C_j is initialized to zero. The channel order of the updated L_C follows the channel order rules.

Figure 4.1(a) illustrates an example of the CCL update with a PCL. In the figure, channel 3 in L_C before the update is removed because it is unavailable in L_P. Furthermore, channel 8 in L_P is added to the CCL because it is a newly available channel that may also be available to neighbors. However, the weight associated with this channel is set to 0. This is represented by the box in white (no shade) that contains channel 8. Finally, channels in the CCL are sorted according to the preference in L_P.

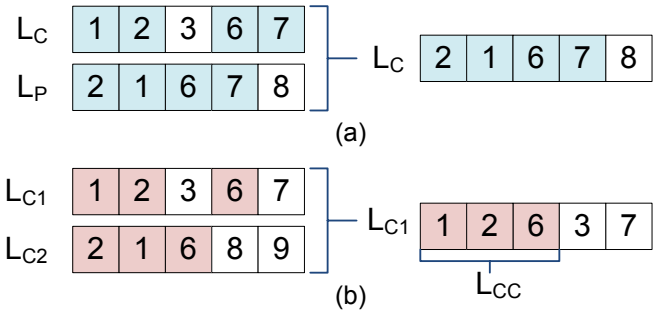


Figure 4.1: Examples of Common Channel List Update (a) CCL Update with PCL (b) CCL Update with Neighbor's CCL.

4.3.2.2 CCL Update with Neighbor's Information

In addition to updating their CCL with sensing information, CR users update their CCL when they receive a CCL from a neighbor. The update is required for the following two purposes: (i) the update determines a list of common channels shared with neighbors. (ii) the information of neighbors' common channel preference can be collected and combined by each CR user for dissemination. Thus, the updated CCL reflects new preference of common channels in the neighborhood.

When a CR user i updates its CCL L_C with its broadcast CCL L_{Ci} and neighbor k's CCL L_{Ck}, the CR user first generates a list of common channels from L_{Ci} and L_{Ck} as follows:

$$L_{CC} \leftarrow L_{Ci} \cap L_{C_k}. \tag{4.7}$$

For each C_j in L_{CC}, the corresponding weight in L_C is set for the neighbor k.

$$L_C \leftarrow \{C_j | W_j : w_{jm} = 1 \text{ for } C_j \in L_{CC}\} \tag{4.8}$$

where $W_j = \sum_{k=1}^{K} w_{jk}$ and K is the number of neighbors sharing the channel C_j in their CCL. As in previous case, the order of L_C follows the channel order rules.

Figure 4.1(b) illustrates the CCL update with a CCL from neighbor 2. As shown in the figure, the common channels of two CCLs are channel 1, 2, and 6. The weights associated with neighbor 2 are set accordingly. Since channel 3 and 7 are unavailable

to neighbor 2, their weight remains 0. Thus, the resulting channel order reflects the new weights in the CCL.

The common channel list update is listed in Algorithm 2. Line 2 to 4 show the addition or the removal of channels for updates with sensing information (PCL). On the other hand, line 7 to 11 outline the updates with neighbor's information (CCL).

Algorithm 4.2 : Common Channel List Update

1: Update with CR user i's Preferred Channel List L_P:
2: $L_P \leftarrow \text{LocalSensing}(\gamma_{pu})$
3: $L_C \leftarrow L_C \setminus \{C_j | C_j \in L_C \text{ and } C_j \notin L_P\}$
4: $L_C \leftarrow L_C \cup \{C_j | C_j \in L_P, C_j \notin L_C, \text{ and } W_j = 0\}$
5:
6: Update with Neighbor k's Common Channel List L_{Ck}:
7: **if** ReceiveBeacon(k, L_{Ck}) from neighbor k **then**
8: $\quad L_{CC} \leftarrow L_{Ci} \cap L_{Ck}$
9: $\quad L_C \leftarrow \{C_j | W_j : w_{jk} = 1 \text{ for } C_j \in L_{CC}\}$
10: $\quad L_C \leftarrow \mathit{OrderChannel}(L_{Ci}, L_{CC}, \gamma_j)$
11: **end if**

4.3.3 Efficient PU Activity Recovery

In this section, we discuss the efficient recovery from the return of a PU to a common control channel. The recovery consists of three steps: new CCC allocation from the common channel list, neighbor list update for lost neighbors, and control radio adaptation for recovering neighbors.

4.3.3.1 Control Channel Allocation

Owing to their dynamic behavior, PUs are highly likely to occupy those established CCCs. Thus, the primary goal is to utilize common channel lists for efficient recovery when PUs are present in the CCCs. When a PU occupies a CCC, CR users tuned to this control channel can immediately detect the change. Without sending any message that may cause interference, CR users choose the best channel in their CCL as the new CCC after the PU-occupied CCC is removed from the list. That is, for $C_j \in L_C$, $Ch_k \leftarrow \min_j C_j$ with $w_{jk} = 1$. Since CR users can reach all or most neighbors by

the new CCC, most neighbors that detect the change in the neighborhood will switch to the new CCC and locate each other by beacon broadcasts. With the exchange of CCLs, most neighbors can be recovered in the new CCC to maintain the network connectivity to the maximum degree.

4.3.3.2 Neighbor List Update

The CCC links associated with each neighbor in the neighbor list show the status of this recovery. If a neighbor k is not yet recovered, the associated CCC Ch_k is a channel no longer available in the CCL. By using this criteria: $Ch_k \notin L_C$, we can adaptively change the operating channel of the control radio to recover neighbors in other common channels. In addition, some existing neighbors may be unable to reach any available channel due to PU activities. In this case, those neighbors should be removed from the neighbor list after having no CCL arrival for a certain period of time.

4.3.3.3 Control Radio Adaptation

A control radio list, denoted by L_R, is a list of channels to which the control radio will be tuned based on the probability of channel selections. If the CCL or neighbor list updates approach a steady state, L_R only includes the allocated CCCs to reduce the switching overhead. In other words, L_R is simply the union of all channels in Ch_k, a small subset of L_C with the best case of single channel, as follows:

$$L_R \leftarrow \cup_{k=1}^{K}\{Ch_k\} \tag{4.9}$$

where K is the number of neighbors. For efficient recovery, the radio list is set to the common channel list when the allocated CCCs no longer exist in the list as follows:

$$L_R \leftarrow L_C \text{ for some } Ch_k \notin L_C. \tag{4.10}$$

Similar to the neighbor discovery process, the probability of selecting the channel from L_R is given in (4.1). Thus, the control radio is tuned to the CCC that reaches

most neighbors with highest probability.

The efficient PU activity recovery is listed in Algorithm 4.3. Line 4 and 5 are new CCC allocations and control radio update in response to PU activities when neighbors can be recovered by the new CCL. Line 8 to 11 show the neighbor list update and control radio adaptation when neighbors cannot be completely recovered by the new CCC. In this case, a neighbor recovery procedure similar to Algorithm 4.1 is required to locate lost neighbors or new ones.

Algorithm 4.3 : Efficient PU Activity Recovery

1: $L_P \leftarrow \text{LocalSensing}(\gamma_{pu})$
2: $L_C \leftarrow \text{UpdateCCL}(L_P)$
3: For neighbor k recovered by new CCL:
4: $Ch_k \leftarrow \min_j C_j \in L_C$ with $w_{jk} = 1$
5: $L_R \leftarrow \cup_{k=1}^{K} \{Ch_k\}$
6:
7: For neighbor k not recovered by new CCL:
8: **if** $Ch_k \notin L_C$ **then**
9: $\quad L_{NB} \leftarrow L_{NB} \setminus \{k\}$
10: $\quad L_R \leftarrow L_C$
11: **end if**
12: $C_j \leftarrow \text{SelectChannel}(L_R)$
13: $\text{SwitchChannelTo}(C_j)$
14: Neighbor discovery as Algorithm 4.1

4.4 Performance Analysis

In this section, we analyze the performance of the proposed scheme by utilizing a mathematical model for delay, throughput, and interference analysis. Moreover, we provide the overhead analysis by comparing our solution with existing grouping and clustering methods, and cosite interference analysis to address the interference issue between the collocated control and data radios.

4.4.1 Analytical Model

To facilitate the performance analysis, we model the CCC recovery and allocation between a neighbor pair as a two-state semi-Markov process. Figure 4.2(a) shows the

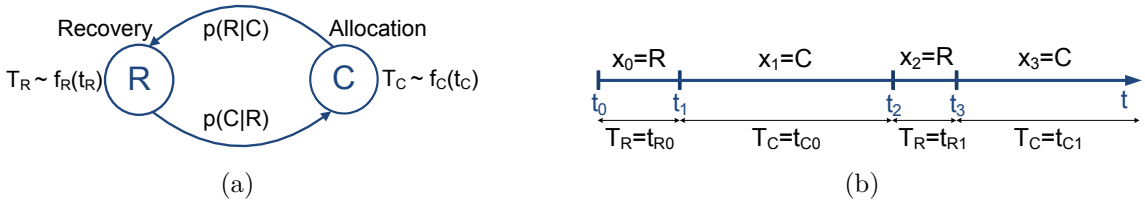


Figure 4.2: (a) Semi-Markov Chain and (b) Alternating Renewal Process for ERCC Performance Analysis.

state diagram of the semi-Markov chain with two states: *Recovery* and *Allocation*. The Recovery state, denoted by R, is the state when CR users are locating neighbors in initial neighbor discovery phase or recovering from the lost of CCC upon PU's return. The Allocation state, denoted by C, is the state when a CCC is allocated to the link between the neighbor pair. The sojourn time in state R, called *CCC recovery time* and denoted by T_R, is a random variable with the distribution $f_R(t_R), t_R > 0$. Similarly, the CCC allocation time T_C is defined as the sojourn time in state C with distribution $f_C(t_C), t_C > 0$. The transition probabilities $p(C|R)$ and $p(R|C)$ are unity in this model. As Figure 4.2(b) shows, by alternatingly staying in each of the two states, the resulting process is essentially an alternating renewal process. For simplicity, we assume that the initial neighbor discovery has the same distribution as other recovery periods.

The average expected recovery time $E[T_R]$ is of great importance because it is the delay of recovering the lost CCC and the indicator of CCC recovery efficiency. To find $E[T_R]$, one needs to determine $f_R(t_R)$. The closed-form expression for $f_R(t_R)$, given in Proposition 4.1, is related to parameters such as PU activities, PU and neighbor locations, PU interference, channel conditions, and number of channels. Here we assume that the CR user u and its neighbor CR user k detect PU correctly and simultaneously when PU changes from inactive to active in channel C_i. We further assume that after C_i is removed from their CCLs, C_j is the only channel in common. If both CR users have more than one channel in common, the probability of meeting

each other on a common channel is higher and the recovery time is smaller. Thus, our assumption is the worst-case scenario.

When the CCLs of the neighbor pair have an identical best channel, say C_j, the recovery is instant. Otherwise, the neighbor pair follows the neighbor discovery procedure and requires the recovery time to meet in C_j. If the probabilities of choosing C_i for CR users u and k are p_1 and p_2, respectively, the probability of CR users meeting on C_i is given by $p = p_1 p_2$. The probabilities p_1 and p_2 can be obtained from (4.1). They are, in general, not identical because L_C or L_R of the neighbor pair is of different length and order. Assume that the success of meeting each other at the m^{th} channel switch is a discrete random variable and denoted by M. If the neighbor pair experiences $m-1$ failures for previous $m-1$ channel switches and succeeds at the m^{th} switch, the probability of successful rendezvous on channel C_i after the m^{th} channel hopping is given by:

$$P(M = m) = (1-p)^{m-1} p. \tag{4.11}$$

This is the PMF of random variable M, which is geometric-distributed. Based on these observations, we obtain the distribution of T_R in Proposition 4.1. Based on the Proposition, one can obtain the average recovery time numerically.

Proposition 4.1 (CCC Recovery Time)**.** *If the CCC recovery time T_R is the sum of M identically, independently, and exponentially distributed random variables with parameter λ, the distribution of CCC recovery time T_R, denoted by $f_R(t_R)$ is given by:*

$$T_R \sim f_R(t_R) = \sum_{m=1}^{\infty} \Gamma(t_R; m, \frac{1}{\lambda}) P(M = m) \tag{4.12}$$

where $\Gamma(t_R; m, \mu)$ is the gamma distribution with shape parameter m and scale parameter μ, and M is a geometric-distributed random variable with the PMF given by (4.11).

Proof. Consider that the CCC recovery time T_R is divided into M intervals, $T_i, 1 \leq i \leq M$, where M is a discrete variable with the PMF given by (4.11) and denotes the number of channel switches required for the neighbor pair to successfully meet each other on one common channel. Assume that the duration of each interval is exponentially distributed with parameter λ, denoted by $T_i \sim Exp(\lambda)$. As a result, T_R is the sum of M exponentially distributed intervals given by $T_R = \sum_i^M T_i$. For each value of $M = m$, T_R given $M = m$ is gamma-distributed with parameters m and $1/\lambda$, denoted by $T_R(M = m) \sim f_R(t_R|M = m) = \Gamma(m, 1/\lambda)$. Therefore, by calculating the joint distribution $f_R(t_R, M) = f_R(t_R|M = m)P(M = m) = \Gamma(m, 1/\lambda)P(M = m)$ and summing up all m's, we obtain the marginal distribution (4.12). □

In practice, the number of channel switches $M = m$ will not be infinite. For large m, the probability $P(M = m)$ is negligible. Thus, the summation in (4.12) starts from 1 to the maximum number of channel switches N_m and $\sum_{m=1}^{N_m} P(M = m) \cong 1$. The resulting distribution is the linear combination of gamma distributions with different parameters m.

For the allocation time T_C, it is mainly determined by PU activities, especially the PUs' arrival rates. This is because once a new CCC is allocated, the allocation will remain mostly unchanged until PU's return to the channel. Even when a CR user decides to change the CCC, the allocation time continues on the new CCC and thus has no state change in this case. Thus, we obtain the distribution of T_C in Proposition 4.2. Based on the Proposition, one can easily calculate the average allocation time as $1/(N_p\beta)$.

Proposition 4.2 (CCC Allocation Time)**.** *Given N_p PUs with the rate of changing from inactive to active β, the distribution of CCC allocation time T_C, denoted by $f_C(t_C)$ is given by:*

$$T_C \sim f_C(t_C) = Exp(N_p\beta). \tag{4.13}$$

Proof. Given N_p inactive PUs on channel C_j, the CR user selects C_j as the CCC and enters the Allocation state. Since each PU arrival follows Poisson distribution with rate β (becoming active with rate β), the arrival rate of N_p PUs is Poisson distributed with $N_P\beta$. As a result, the interarrival time between two PU arrivals is exponentially distributed with parameter $N_p\beta$. Based on the assumption that C_j is available, all active PUs must become inactive before the CR user switches to C_j. Moreover, due to the memoryless property of exponential distribution, the PU inactive time before the CR user switches to C_j is irrelevant. Thus, we obtain the distribution of allocation time, $f_C(t_C)$, as in (4.13). □

4.4.2 Delay, Throughput, and Interference

To find the delay and control throughput, we assume that CR user i transmits control data to a neighbor k on CCC C_j. The maximum achievable rate for the control transmission is given by:

$$R_j^k = B\log(1 + \frac{P_{su}|h_{ik}|^2}{N_0B + \gamma_j^k}) \tag{4.14}$$

where B is channel bandwidth, P_{su} is CR user transmit power, h_{ik} is the channel gain of the link between CR users i and k, $N_0/2$ is the power spectral density of additive white Gaussian noise, and γ_j^k is the accumulated interference power of PU transmit signals observed by CR user k on C_j. If the CR user has N_k neighbors within its transmission range R_s and all neighbors are tuned to C_j, we refer to the area covered by R_s as a *control capacity region*. Since the achievable throughput is limited by the rate of the weakest link, where the interference power γ_j^k is the largest and the channel gain h_{ik} is the smallest, the maximum achievable throughput in the capacity region is given by $R_j = \min\{R_j^k, k = 1, \ldots, N_k\}$. If the control packet is of length L bits, the transmission delay is L/R_j.

Now if N_s CR users are uniformly deployed in the area A, there are approximately $N_r = A/(\pi R_s^2)$ control capacity regions. The CCC may be the same in each region

while the PU interference levels and the channel conditions are different. Thus, the maximum control throughput of the CR ad hoc network, called the sum-rate capacity, can be obtained by the maximum sum of rates from all regions, $R_j(q), q = 1, \ldots, N_r$, as follows:

$$R_c = \max_{R_j(1),\ldots,R_j(N_r)} \sum_{q=1}^{N_r} R_j(q), \quad C_j \in \{1, \ldots, N_c\}. \tag{4.15}$$

ERCC intelligently selects the CCC C_j in each region such that the sum-rate capacity (4.15) is achieved.

If a PU is active on channel C_j and the area covered by PU's transmission range R_p is called the *protected region*, the interference with PUs only results from those transmitting CR users outside the protected region. Since there are only $N_r^{'} = (A - \pi R_p^2)/(\pi R_s^2)$ possible control capacity regions and one transmitting CR user in each region, the maximum accumulated interference from those CR users observed by the PU on C_j is:

$$P_I = K \sum_{i=1}^{N_r^{'}} P_{su} \Big(\frac{d_0}{d_{pi}}\Big)^{\delta}, \quad d_{pi} > R_p, \forall i, \tag{4.16}$$

where K is an antenna-related constant, P_{su} is the CR user transmit power, d_0 is the reference distance, d_{pi} is the distance between the PU and the CR user i, and δ is the path loss exponent.

4.4.3 Overhead

The overhead of the ERCC algorithm is dominated by regular broadcasts of maintained CCLs. The frequency of the broadcasts determines the accuracy of channel conditions in the CCLs and the overhead incurred by ERCC. Thus, the choice of the broadcast rate is essential for reducing the unnecessary overhead.

The CCL updates and broadcasts are related to PU activities and the broadcast rate of neighbors, since CCLs are updated with i) the PCL from the sensing results and ii) the CCL received from neighbors. Assume that a CR user has N_k neighbors and neighbor k, $k = 1, \ldots, N_k$, broadcasts its CCL with the rate r_k. The death

and birth rates of PU model are α and β, respectively. Since the periodic sensing frequency r_s must be no less than the PU activity change rate and CR users need to broadcast CCLs with new channel conditions, we assume that $r_s = \max\{\alpha, \beta\}$ where $\alpha, \beta \leq r_k, \forall k$. Thus, the rule of thumb for choosing the broadcast rate r_i is formulated as follows:

$$r_s = \max\{\alpha, \beta\} \leq r_i \leq \min\{r_1, \ldots r_{N_k}\}. \tag{4.17}$$

By adaptively selecting the broadcast rate based on (4.17), the CR user and its neighbors broadcast CCLs with the rates gradually approaching the PU activity change rate α or β, whichever is the largest, to minimize the unnecessary broadcasts.

4.4.4 Cosite Interference

As described in Section 4.2, each CR user is equipped with two radios dedicated to control and data channels, respectively. Owing to their collocation in each CR user, the out-of-band (OOB) emission [105] from a transmitting radio (control or data) can block the transmission or corrupt the reception at the other radio operating in a different channel within the same band [41]. This phenomena, called *cosite interference*, results in degraded performance: an unreliable CCC and data transmissions with compromised data rate.

Contrary to the setting in [41] where all radios are used for data transmission, ERCC utilizes separate radios for control and data. The key differences in ERCC are twofold: i) to ensure the reliability of the CR ad hoc network, any operation in a CCC has higher priority than those in data channels, and ii) since all channels have the same bandwidth, CCC transmission is short compared to data transmissions. Thus, medium access control (MAC) techniques [105] such as prioritized time sharing, power control, and dynamic channel allocation can be utilized to ensure the reliability of CCC and mitigate the cosite interference while the throughput of the data channel is not considerably compromised.

4.4.4.1 Prioritized Time Sharing

When both control and data radios are transmitting, only one radio is active at a given time [41]. Owing to its high priority, the control radio temporarily refrains the data radio from transmitting in data channels whenever the control radio transmits. The throughput of data transmission is only slightly for short control traffic. Similarly, the data radio temporarily ceases transmitting whenever the control radio starts receiving control data. If the control radio transmits when the data radio receives data from others, the control radio notifies the transmitting neighbor to temporarily stop the data transmission. Moreover, the CR users can broadcast their regular control traffic schedule to neighbors so that their schedulers automatically cease data transmission during the scheduled control traffic period.

4.4.4.2 Power Control and Rate Adaptation

To reduce the power leakage from the receive data channel to the control channel, the control radio notifies the transmitting neighbor to perform transmission power control and adjust the transmission rate in the data channel for cosite interference mitigation.

4.4.4.3 Dynamic Channel Allocation

The data channel can be dynamically reallocated to the one far separated from the CCC, if possible, to further reduce the cosite interference. The control channel can also be dynamically changed if the CCC quality degrades.

4.5 Performance Metrics

The performance of CCC establishment can be evaluated in a variety of ways. For the convenience of performance evaluation, we define four metrics in this section: CCC link indicator, CCC coverage indicator, best channel indicator, and PU interference as follows:

4.5.1 CCC Link Indicator

A link is said to be available between two CR users if they are located within their transmission range and observe at least one channel in common, but have not located each other in any channel. When neighboring nodes operate and exchange information in a common channel, a CCC link is established. Thus, we define a CCC link indicator (CLI) as the percentage of established CCC links over all available links in the network as follows:

$$CLI = \frac{N_{disc}}{N_{tot}} \tag{4.18}$$

where N_{disc} is the number of current established CCC links and N_{tot} is the number of total available links. Since topology and neighbors change as PU activity changes, the CLI also indicates how fast CR users establish links with new neighbors and recover links with old neighbors. Moreover, the CLI value achieves unity when all neighbors are discovered and CCC links are established. Therefore, this indicator is an alternative way of evaluating neighbor discovery rate and the responsiveness to PU activities.

4.5.2 CCC Coverage Indicator

The coverage of a CCC refers to an area covered by links allocated to a control channel. Since obtaining an exact footprint of those links is nontrivial, a CCC coverage indicator (CCI) is used as an alternative to evaluate the coverage of CCC distribution in the network. The CCC distribution refers to the number of CCC links distributed in all licensed channels. Thus, the CCC coverage indicator is defined as follows:

$$CCI = \frac{STD(p_{dist})}{STD(p_{best})} \tag{4.19}$$

where $STD(p_{dist})$ is the standard deviation (STD) of current CCC distribution over all licensed channels and $STD(p_{best})$ is the STD of the CCC distribution in the best case. If p_i is the number of CCC links in channel C_i, $\overline{p}$ is the average number of

all p_i's, and N_c is the number of licensed channels, the standard deviation of CCC distribution p is defined as $STD(p) = \sqrt{\frac{1}{N_c}\sum_{k=1}^{N_c}(p_i - \overline{p})^2}$. For different number of licensed channels or available links, standard deviation of the distribution can vary significantly. Thus, for easy comparison of different test cases, $STD(p)$ can be normalized by the STD of CCC distribution in the best scenario. The best case can be achieved when all CR users use a single CCC. That is, $p_i = N_{tot}$ and $p_j = 0$ for $i \in \{1, \ldots, N_c\}$, $1 \leq j \leq N_c$ and $i \neq j$. Evidently, the STD of CCC distribution in the best scenario is the maximum for a given number of channels N_c. Therefore, the CCI indicates how close current CCC distribution to the distribution in the best case and the CCI value achieves unity when all CCC links in the network are established in the same channel.

4.5.3 Best Channel Indicator

The best channel indicator (BCI) indicates the percentage of CCC links to which the best quality channel observed at each CR user is allocated. Thus, the BCI is defined as follows:

$$BCI = \frac{N_{best}}{N_{tot}} \tag{4.20}$$

where N_{best} is the number of CCC links to which the best quality channel in the PCL is allocated and N_{tot} is the number of total available links. The BCI value achieves unity when all the CCC links are of the highest observed channel quality.

4.5.4 PU Interference Indicator

The PU interference (PUI) indicates the average accumulated PU transmit signal power observed on channel C_j at each CR user when C_j is used for control transmission, and is given by:

$$PUI = \frac{\sum_{i=1}^{N_s} \gamma_i^j}{N_s}, \quad C_j \in \{1, \ldots, N_c\}, \tag{4.21}$$

where N_s is the number of CR users, N_c is the number of channels, and γ_i^j is the accumulated interference power on the control channel C_j at CR user i. Since higher PU interference level implies a higher possibility of PUs in the surrounding area, this metric indicates the average level of interference with PUs per CR user during control transmission. Moreover, due to the interference at the CR user, this metric can also be used to evaluate the level of achievable control throughput. In general, the higher the PUI level, the lower the control throughput.

In the next section, we evaluate the performance of our proposed method with the metrics defined in this section.

4.6 Performance Evaluation

In this section, we discuss the simulation setups and evaluate the performance of our proposed ERCC method in several test scenarios. We first introduce our simulation environment, compare the analytical model with the simulation model, and then describe seven test cases for performance evaluation.

4.6.1 Simulation Environment

In our simulation environment, we assume that a number of CR users are randomly deployed in a square area $500\,\mathrm{m} \times 500\,\mathrm{m}$ sharing a set of licensed channels with PUs in the 5.2 GHz frequency band. Both PU and CR user transmit powers are set to 0.1 W. The PU and CR user interference thresholds are set to γ_{pu} and γ_{su}. These settings correspond to PU and CR user transmission ranges R_p and R_s, respectively. For example, for $\gamma_{pu} = -72.7\,\mathrm{dBm}$, $\gamma_{su} = -66.7\,\mathrm{dBm}$ and wavelength $\lambda = 0.058$, the PU and CR user transmission ranges are approximately 200 m and 100 m, respectively. The noise floor is set at -101 dBm. For correlated shadowing, the decaying coefficient a in the exponential correlation model is set to 0.002 for suburban settings [32]. This corresponds to the decorrelation distance of approximately 346 m where the correlation drops to 0.5 and ensures that the observations of neighbors are highly

correlated ($\rho_{ij} > 0.8$). For convenience, the number of PUs, CR users, and licensed channels are denoted by N_p, N_s, and N_c, respectively. In addition, the PU density, PU ON/OFF period, PU transmission range, CR user transmission range, and log-normal shadowing dB spread are denoted by D_p, t_p, R_p, R_s, and σ_{dB}, respectively.

For performance comparison, we select a group-based CCC design approach from [18], denoted by *GRP*, and a sequence-based approach from [23], denoted by *SEQ*, as references. These two selected reference approaches are summarized as follows:

- *GRP:* CR users exchange quantized channel quality information by sending Hello messages to neighbors. Based on the channel quality values received from neighbors, CR users adaptively update a probability list for control channel selection. The probability for a channel is higher if more neighbors select that channel as the control channel. The channel with the highest probability is selected as the common control channel. Thus, control channels are selected according to the decisions of the majority of neighbors. The settings used in GRP are: A = 0.1, B = 1.5, and C = 4 for probability list update. The number of quantized receive power levels for determining quality values is 128.

- *SEQ:* Each CR user constructs a channel hopping sequence by using permutations of available channels. A neighboring CR user pair establishes a control link after both CR users hop to the same channel and exchange information. To establish other control links, both CR users hop to other channels based on their own sequence. If the channel is occupied by a PU, the channel is removed from the hopping sequence. New sequence is generated for new channel availability obtained from local sensing information.

PU activities follow the two-state birth-death process with the birth rate 0.3 and the death rate 0.2. In this case, PUs fix their location and operating channels, but may be active or inactive based on the state of the process. When a PU is inactive,

the PU-occupied channel is considered free until the PU is active. The degree of PU activities is determined by the ON/OFF period t_p.

The observation time for each topology is set for 10 minutes. For neighbor discovery and message exchange, each CR user is tuned to a channel for 200 ms. During the time interval, CR users perform local spectrum sensing, broadcast channel and neighbor information, determine new common channel lists, and allocate available channels to CCCs accordingly. The metrics are collected every 200 ms after SEQ changes its hopping channel. All results are averaged over the observation time and 10 random network topologies, in which PUs and CR users are uniformly distributed in the deployment area. Although the synchronization of CR users is not required, all nodes are simultaneously activated in the test cases.

4.6.2 Comparison of Analytical and Simulation Models

To compare the analytical model introduced in Section 4.4.1 with the simulation model, we focus on a neighboring CR user pair and their average CCC recovery time. In the analytical model, the average recovery time is obtained by calculating the expected recovery time numerically using the distribution from (4.12) with the maximum channel switches N_m set to 50. In the simulation model, the recovery time of a neighboring pair is averaged over all occurrences of CCC recovery during the entire observation time and the random network topologies under testing.

Figure 4.3 shows the comparison of the average recovery time from the analytical model and the simulation model under various degrees of PU activity characterized by the probability of PU ON state, P_{on}. In general, the CCC recovery time is linearly increased with the number of available channels. This is because the CR user may choose other available channels not common to the neighbor of interest, which results in the increase of the average recovery time, even though the probability of choosing

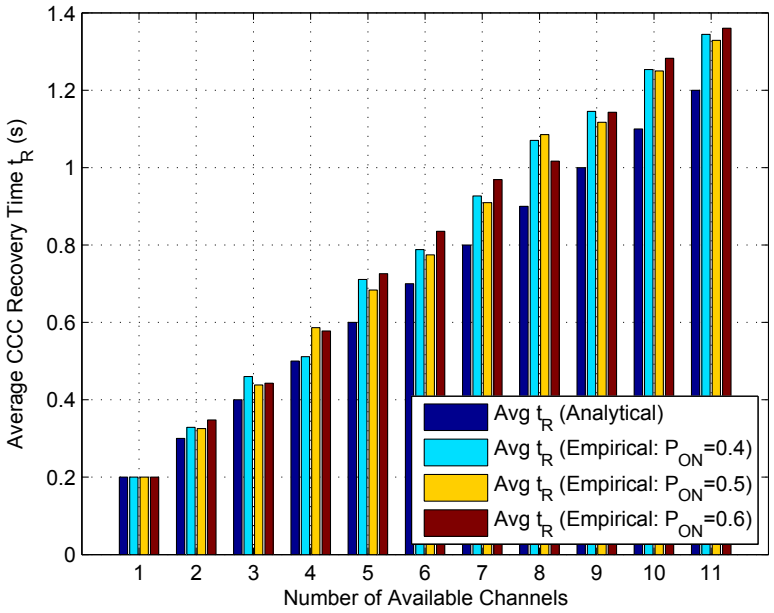


Figure 4.3: Comparison of Average CCC Recovery Time in Analytical and Simulation Models.

the common channel in the CCL is the largest. In addition, given a number of available channels, the average recovery time does not vary significantly under different levels of PU activities. Although PU activity affects the channel availability and the probability of channel selections, the recovery time is dominated by the number of channel switches once the available channels are determined in the CCL. Thus, the probability of selecting the common channel (p in (4.11)) for recovery remains constant if there is no CCL update due to PU activities. More importantly, the figure shows that the empirical values from the simulation model closely follow the analytical values as the number of channels varies. Therefore, the analytical model provides the first-order analysis and prediction of the average CCC recovery time.

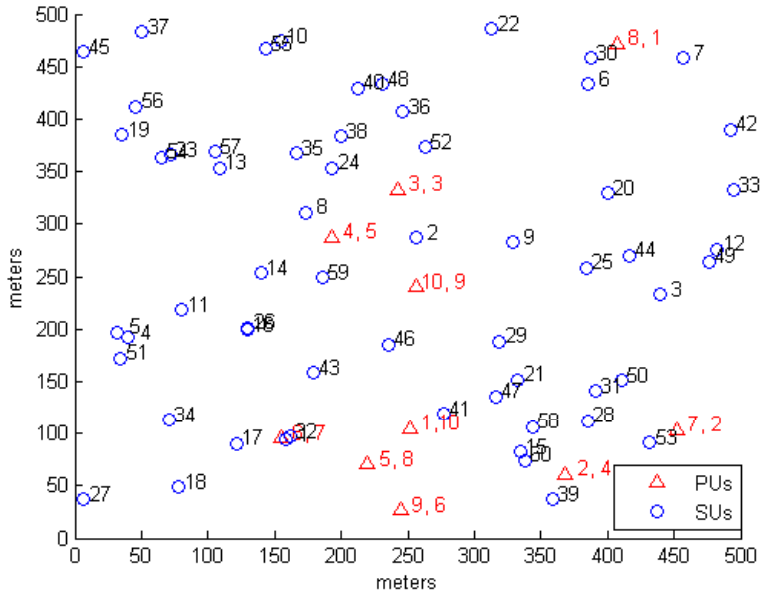
4.6.3 Test Cases

To evaluate the performance, we test our proposed ERCC solution in the following test cases: (i) neighbor discovery, (ii) PU ON/OFF period, (iii) PU transmission range, (iv) PU density as the number of PUs per channel, (v) CR user transmission range,

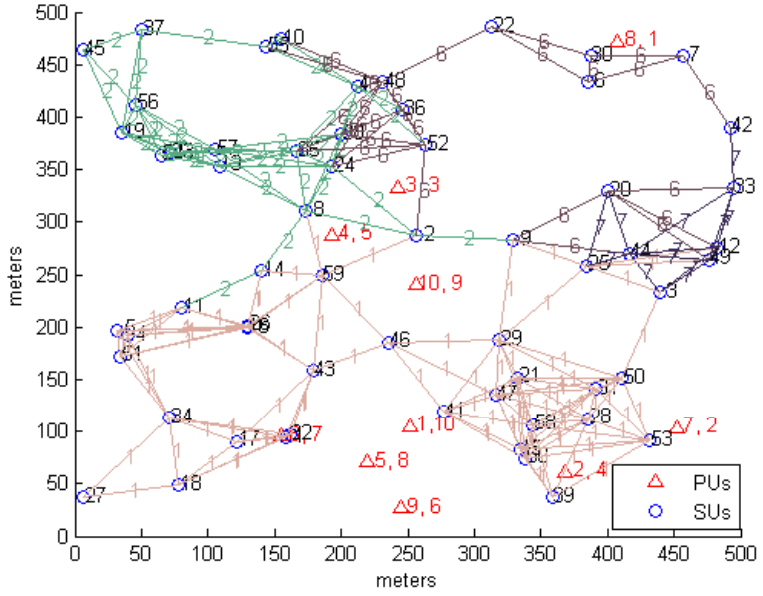
(vi) the scalability or the density of CR user population, and (vii) shadow fading for a range of dB spread. These test cases will show how our solution performs under the impacts of PU activity, network topology changes, and channel impairments. The configuration used in each test case for cross reference is $N_p = 10$, $N_s = 60$, $N_c = 10$, $D_p = 1$, $t_p = 4\,\text{s}$, $R_p = 200\,\text{m}$, $R_s = 100\,\text{m}$, and $\sigma_{dB} = 0\,\text{dB}$. In each test case, we evaluate the performance of all three methods by varying one of the parameters and illustrate the average and standard deviation of metric values versus the parameter of interest. In the figures of this section, the top left, top right, bottom left, and bottom right sub-figures show the CCC links, CCC coverage, best channel, and PU interference metric values, respectively.

4.6.4 Neighbor Discovery

We first demonstrate the performance of ERCC neighbor discovery algorithm and the network topology achieved by neighbor discovery. Figure 4.4(a) shows an example of initial deployment of a CR ad hoc network overlaid with a primary network. The primary network consists of 10 PUs, represented by red triangles. Each of which occupies one of 10 licensed channels. The numbers near each triangle are PU's ID followed by its operating channel. The CR ad hoc network, represented by blue circles, consists of 60 CR users. The number next to a circle is a CR user ID. The available links, not shown in this figure, depend on the channel availability of each neighboring pair. After the CR users in the network start the neighbor discovery and CCC allocation, the network topology can be established in a short period of time. Figure 4.4(b) illustrates the network connectivity with full neighbor discovery at time unit 37. Each colored line between two CR users represents an established link between a neighboring pair. The number in the middle of each line shows the channel allocated as the CCC. As shown in the figure, more than half of the links in



(a)



(b)

Figure 4.4: (a) Initial Deployment at $t = 0$ and (b) Network Topology with Full Neighbor Discovery at $t = 37$ ($N_p = 10$, $N_s = 60$, $N_c = 10$).

the bottom use channel 1 as the CCC because the PU occupies channel 1 in the up-right corner. The rest of the links share three other control channels in the network due to different observed channel availability. One global CCC is infeasible in this case because each PU occupies one licensed channel.

4.6.4.1 Primary User ON/OFF Period

PU ON/OFF period is the smallest duration of a PU being active or inactive. Based on the state in the birth-death process, a PU may be consecutively active or inactive for several periods. During these periods, PU activities can be considered stationary. Thus, increasing the period reduces the frequency of dynamic changes in PU activity. Figure 4.5 shows the four expected metric values of three methods under testing in the range of PU ON/OFF period from 0.2 to 8 seconds. As shown in the figure, ERCC steadily improves the connectivity with neighbors, increases the CCC coverage, and selects more channels of the best quality while maintaining the lowest interference with PUs among all three methods, as PU activities appear to be less dynamic on average. This proves its capability of efficient recovery from high PU activities. Specifically, ERCC maintains *at least* 80% of CCC links when PU activities are most dynamic and also improves its connectivity to almost 100% when the activity is less intense while GRP can only achieve *at most* 80% of connectivity. Even though GRP has better CCC coverage than ERCC under highly dynamic PU activities, it is achieved at the expense of causing more interference. Moreover, SEQ appears to be less susceptible to PU active periods. However, it achieves low indicator values and causes more interference than ERCC because SEQ selects channels for CCC links based on hopping sequences constructed with no consideration of channel quality and neighbor information. Thus, ERCC makes better tradeoffs between increasing coverage and choosing a channel of best quality for minimizing the interference.

4.6.4.2 Primary User Transmission Range

The PU transmission range is determined by the path loss model with specified PU transmit power and the receive PU signal threshold. We can change transmit power to obtain different transmission ranges. Alternatively, with the fixed transmit power, we assume that CR users change the thresholds for different levels of tolerable PU

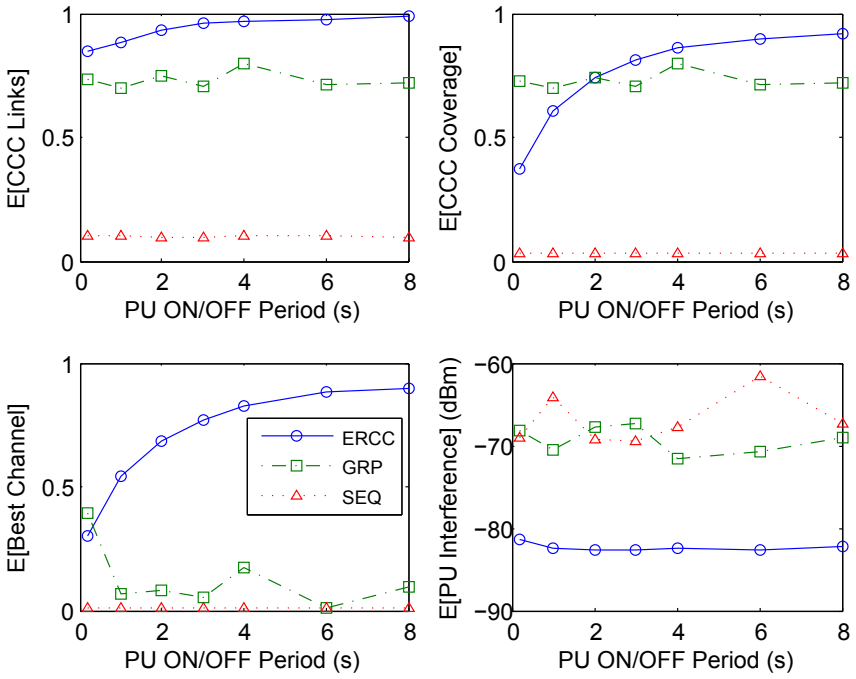


Figure 4.5: Expected Metric Values vs. PU ON/OFF Period t_p.

interference and PU transmission range. The larger the PU transmission range is, the more homogeneous the spectrum availability is in a neighborhood. Figure 4.6 shows the expected metrics from the PU transmission range 100 to 500 m. These ranges correspond to PU threshold γ_{pu} values from -92.72 to -110.70 dB. As shown in the figure, ERCC utilizes the local spectrum homogeneity to improve all metrics as the PU transmission range increases. For the same reason, SEQ slightly improves its performance. As the range increases, the hopping sequences chosen by neighbors in SEQ are more similar for better chances of rendezvous. Conversely, the performance of GRP drops significantly as the range increases. This is because as the range of the PU on each channel increases, the probabilities for selecting control channels in GRP appear to be more comparable. As a result, CR users using GRP in a neighborhood cannot easily agree upon their control channel selection. This test case shows that ERCC is more consistent and reliable than the other two methods as PU adapts its transmit power and range.

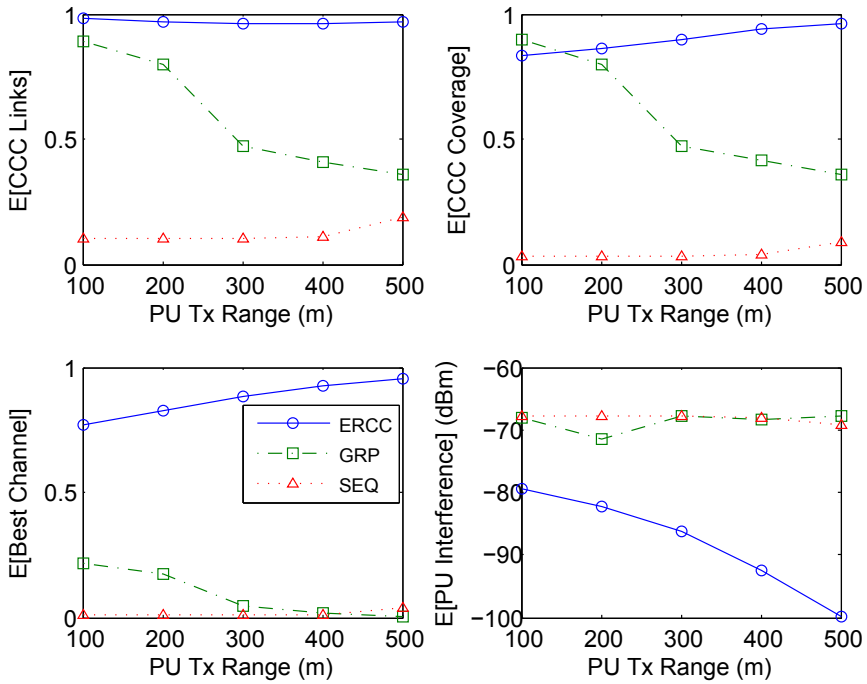


Figure 4.6: Expected Metric Values vs. PU Transmission Range R_p.

4.6.4.3 Primary User Density

In this test case, we increase the PU density by increasing the number of PUs per licensed channel within the testing area. This will increase the observed PU interference level and reduce observed channel quality if more than one PU is active. Figure 4.7 shows expected metric values versus one, two, and three PUs occupying each licensed channel. As expected, the interference level increases for all methods as the density of PUs increases. Even though channel quality deteriorates, ERCC maintains high percentage of links with neighbors and best channel selections partially based on the ordering of the channel quality. On the contrary, the performance of GRP is considerably affected by the reduced channel quality, since it updates channel selection probabilities with channel quality values. As in previous cases, SEQ does not achieve high coverage and connectivity as the other two methods, even though it is insensitive to PU parameter changes. Therefore, these results show that ERCC is capable of adapting to high interference environment with minor coverage reduction.

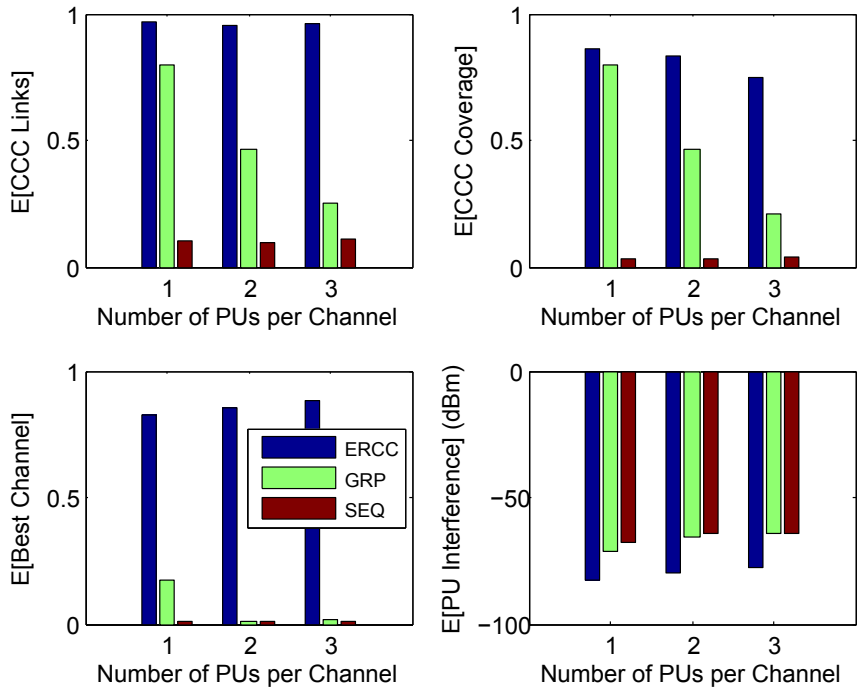


Figure 4.7: Expected Metric Values vs. Number of PUs per Channel D_p.

4.6.4.4 Secondary User Transmission Range

Similar to the *PU transmission range* test case, we vary the CR user transmission range by changing the CR user sensing threshold γ_{su}. As the range increases, more neighbors are covered, resulting in increased number of neighbors. Figure 4.8 shows expected metrics versus CR user transmission ranges from 50 to 250 m. In general, the performance of both ERCC and GRP slightly decreases as the CR user transmission range increases. Since more neighbors away from the neighborhood contribute to the message exchange, the channel list may not reflect the real channel conditions in the surrounding area. Interestingly, the performance of GRP also degrades when the range is small. Since CCC allocation in GRP relies on the updates from the majority of neighbors in the neighborhood, small CR user range covers only a few neighbors that may not represent the real majority of neighbors for correct channel selection. This test case shows that few benefits can be obtained from increasing CR user range

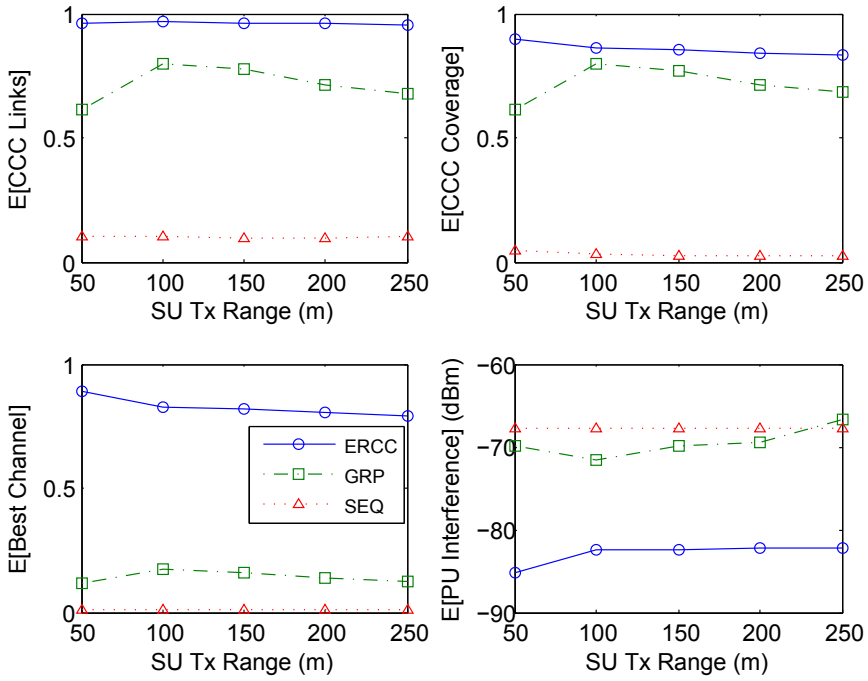


Figure 4.8: Expected Metric Values vs. CR User Transmission Range R_s.

to a large value, not to mention the waste of transmit power and higher interference incurred. However, proper transmission range is still essential to methods such as GRP for achieving good performance.

4.6.4.5 Scalability of CR user deployment

The scalability of CR user deployment is evaluated by varying the number of CR users in the testing area. This also changes the density of CR user population in the fixed area. Similar to the *CR user transmission range* test case, the change of CR user density affects the number of neighbors in the neighborhood. Figure 4.9 illustrates the expected metric values versus the number of CR users ranging from 30 to 150. As shown in the figure, the performance of ERCC and SEQ is consistent and thus scalable in the range under testing. GRP, in general, is also scalable. However, too many or too few neighbors degrades its performance. Thus, GRP is more sensitive to CR user parameters and neighbor updates while ERCC and SEQ exhibit the scalability for a variety of different CR user deployment sizes.

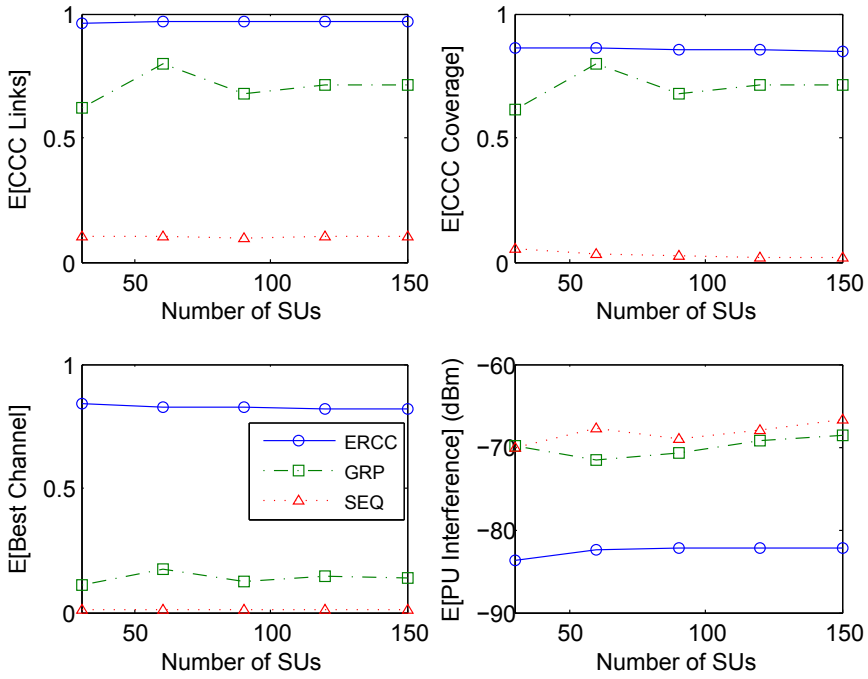


Figure 4.9: Expected Metric Values vs. Number of CR Users in Deployment N_s.

4.6.4.6 Shadow Fading

Unlike all previous test cases that PU signal quality is deteriorated only by path loss model, this test case evaluates the performance of CCC solutions with the addition of independent and correlated shadow fading to reflect more realistic channel conditions. With the increase of the log-normal shadowing dB spread σ_{dB} in the channels, the received PU signal power varies so greatly that the CR users are more susceptible to incorrect detection of PUs and channel availability. In this test case, we assume that all packets for message exchange between neighbors are protected by upper layer error control schemes and received correctly.

Figure 4.10 shows the expected metrics versus the dB spread values in both independent and correlated shadow fading. ERCC outperforms GRP and SEQ in terms of all the metrics. However, the performance of ERCC and GRP gradually degrades in independent shadowing as σ_{dB} increases. Unlike ERCC and GRP, SEQ is less susceptible to σ_{dB} changes. Interestingly, ERCC maintains better CCC links and

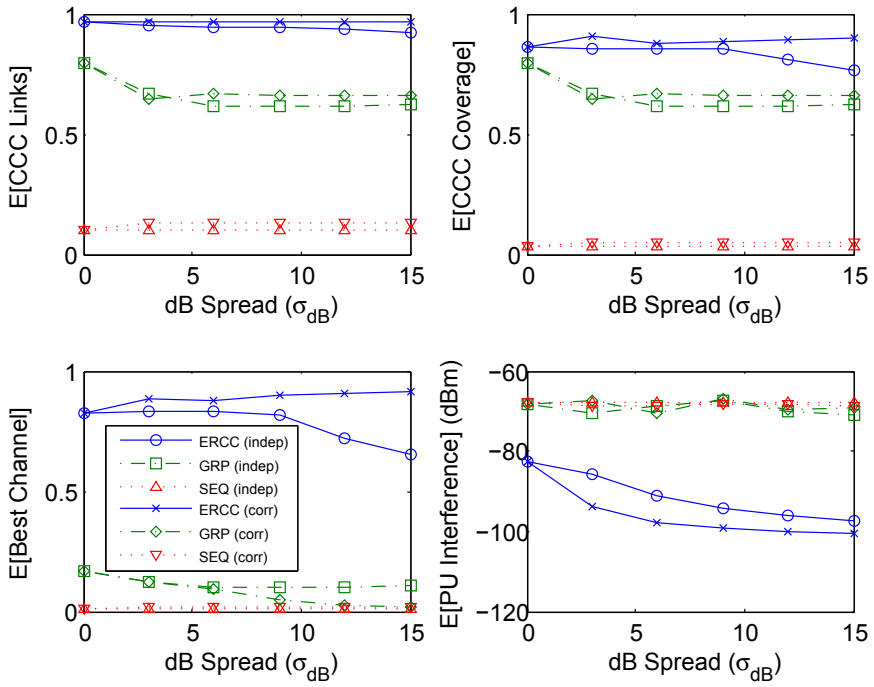


Figure 4.10: Expected Metric Values in Shadow Fading σ_{dB}.

coverage in correlated shadowing than those in the independent case. This is because when the neighbors' observations are correlated, their CCLs tend to be similar even with large dB-spreads, which facilitates the CCC allocation and improves the CCC coverage in a deep shadow. However, due to inaccurate received PU power levels, it is possible to incur the interference with PUs in this case. Thus, any cooperative spectrum sensing scheme [13, 32] can be incorporated into ERCC to mitigate the effects of channel impairment. By using the established CCC links among neighbors, neighboring CR users in ERCC can exchange spectrum sensing information to improve the detection of PUs and obtain fading-independent CCLs for robust CCC establishment and better CCC coverage.

CHAPTER V

REINFORCEMENT LEARNING FOR COOPERATIVE SENSING GAIN

5.1 Motivation

In Chapter 3, we discuss cooperative gain and cooperation overhead in cooperative sensing. We know that, regardless of the benefits of cooperative sensing, cooperation incurs overhead such as (i) shadowing correlation, (ii) control message overhead, (iii) synchronization and reporting delay, and (iv) user and data reliability that limits the cooperative gain. First, it is known that shadowing correlation degrades the performance of cooperative sensing [32]. This is because CR users, spatially located in proximity and blocked by the same obstacle, may experience correlated shadowing and have poor observations of PU signals. As a result, cooperative gain is limited by shadowing correlation. Second, cooperation requires extra control message exchange among CR users for reporting sensing data on a CCC [54,57]. Such control transmission is also limited by the available CCC bandwidth. Third, synchronizing CR users in CR ad hoc networks for sensing cooperation is not a trivial task. Since CR users have different transmission and sensing schedules, the local sensing results from cooperating CR users may not simultaneously arrive at the FC. Moreover, control packet collision and re-transmission in control channel result in extra reporting delay. Thus, asynchronous reporting and delay overhead should be considered in cooperative sensing. Finally, the reported sensing results may be unreliable due to the malfunctioning of CR users, or manipulation of malicious CR users, known as the Byzantine failure problem [17]. Furthermore, control channel fading incurs reporting errors, which may

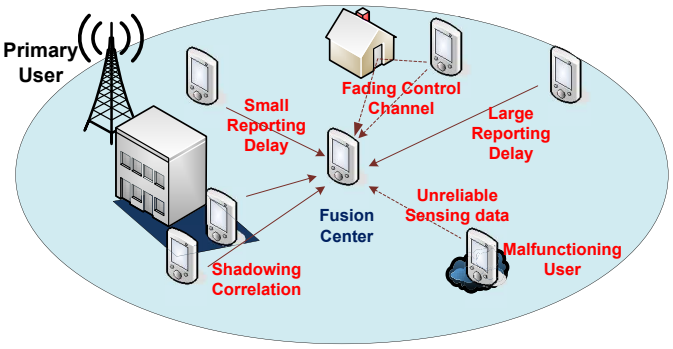


Figure 5.1: Cooperative Sensing and Possible Cooperation Overhead that Limits Cooperative Gain.

further complicate the reliability issue. Therefore, cooperative sensing needs a mechanism that excludes unreliable cooperating users as well as their sensing results from cooperation. Figure 5.1 illustrates an example of cooperative sensing and possibly incurred cooperative overhead in a CR ad hoc network.

Existing cooperative sensing solutions are mainly based on the model of parallel fusion network in distributed detection [90], where all cooperating CR users generate local decisions and report them simultaneously to FC for making global decisions by data fusion. To mitigate correlated shadowing, [89] takes into account user correlation in the linear-quadratic fusion method to improve detection performance in correlated environment. In addition, [78] proposes user selection algorithms based on location information to find uncorrelated users for cooperative sensing. However, these solutions may not be able to adapt to dynamic environmental changes in a timely fashion. To reduce control messages overhead, [59,91,92,104] report quantized and binary sensing data for soft and hard decision combining, respectively. Alternatively, [84] reduces the average number of reporting bits by restraining unreliable sensing results from being reported. For synchronization and delay issues, recent studies [82, 100, 104] consider the asynchronous case where cooperating CR users report local results at different times. However, conventional schemes based on the parallel fusion model [17, 89, 92] typically assume that observations among CR users are conditional independent, and

CR users are perfectly synchronized with instant reporting on an error-free CCC. Moreover, existing cooperative sensing methods seldom address all aforementioned cooperation overheads in response to dynamic environmental changes. Thus, it is clear that a new model for cooperative sensing with the capability of interacting with and learning from the environment is required to tackle all these problems in CR ad hoc networks.

In this Chapter, we introduce a novel reinforcement learning-based cooperative sensing (RLCS) method to address incurred cooperation overheads and improve detection performance in multipath and correlated shadow fading. Reinforcement learning (RL) [85] is an adaptive method for a decision-making agent learning to choose optimal actions and maximize received rewards by interacting with its environment. In RLCS, the CR user acting as the FC is the decision-making agent interacting with the environment that consists of its cooperating neighbors and their observations of PU activities. By requesting sensing results from its neighbors, the FC learns the behaviors of cooperating CR users and takes actions to select users for cooperation through periodic cooperative sensing. Among a variety of RL algorithms, temporal-difference (TD) learning [85] is utilized in this work to address cooperation overhead issues in cooperative sensing. It is suitable for mitigating cooperation overheads due to its capability of evaluating the correlation between successive CR user selections, adjusting subsequent selection predictions based on the experiences accumulated over time, and selecting an optimal set of cooperating CR users. More importantly, TD learning enables CR users to learn from and adapt to dynamic environmental changes, such as the changes in PU activities, user movement, user reliability, and channel conditions, while maintaining satisfactory sensing performance without requiring a model or a priori knowledge about PU activities and CR users' behavior. Obviously, these benefits of TD learning cannot be obtained by pre-selection of CR users or cooperation

of all CR users with no learning. Although RL algorithms have been applied to dynamic channel access [7,24], user selections [93], and multi-band sensing policy [66,67] in CR networks, to the best of our knowledge, RLCS is the first work applying the RL techniques to address both the cooperation overhead problems and the detection performance of cooperative sensing. Therefore, in Section 5.2, we present our RLCS system model and assumptions. In Section 5.3, we introduce the RLCS scheme including cooperative sensing decision process and RLCS algorithm. In Section 5.4, we discuss RLCS performance analysis including optimal solution and convergence of RLCS algorithm, optimal stopping time for fast response, and the impact of fading control channels. In Section 5.5, we evaluate RLCS detection performance and its adaptability to environmental change by test scenarios and numerical results. The variables and notations used in this Chapter are tabulated in Table A.3 and A.4 of Appendix A for reference. Our contributions are summarized as follows:

- We propose the novel RLCS model and algorithm for CR users to learn the optimal user selection policy for finding uncorrelated and reliable cooperating neighbors to improve cooperative gain and mitigate cooperation overhead in cooperative sensing.

- We show that the optimal solution obtained by RLCS approach greatly improves the detection performance under correlated shadowing while minimizing control channel bandwidth requirement by using binary local decisions and hard-combining strategy.

- We demonstrate that RLCS converges asymptotically with the option of optimal stopping for fast response in dynamic environment, mitigates the impact of control channel fading, improves the reliability of user and sensing data selection, and adapts to PU activity change and the movement of CR users.

5.2 RLCS System Model

We consider a group of CR users forming a CR ad hoc network overlaid with a primary network to opportunistically share a set of N_C licensed channels. Each licensed channel is assumed to be occupied by one primary transmitter (the PU) and potential primary receivers in its transmission range. In order to protect these primary receivers from interference, the range of PU transmission R_P plus the range of CR user transmission R_S, $R_P \gg R_S$, forms the protected region [89]. The PU activity on channel m is modeled as a two-state birth-death process with the birth rate r_b^m and the death rate r_d^m [49]. In this PU model, the transition follows a Poisson process with exponentially distributed inter-arrival time. Thus, the long-term average probabilities of PU active (P_{on}^m) and inactive (P_{off}^m) on channel m are $\frac{r_b^m}{r_b^m+r_d^m}$ and $\frac{r_d^m}{r_b^m+r_d^m}$, respectively. The PU activity is unknown to CR users a priori. To balance the traffic load and power consumption, the CR users in the CR ad hoc network may either take turns to serve as the FC to cooperatively sense one licensed channel each time, or act as FCs simultaneously to sense multiple channels at the same time. However, there is only one CR user acting as the the decision-making FC (learning agent) on each channel. Without loss of generality, we focus on RLCS with one FC and its cooperating neighbors on one channel and the channel index m will be omitted from the notation thereafter unless otherwise specified. How to determine which channel to sense is beyond the scope of this work.

Let $\mathcal{C}$ be the set of the neighbors of the FC where the FC is denoted by CR user 0 and $|\mathcal{C}| = L$. Let y_i be the average SNR in dBm of the received PU signal observed at cooperating CR user i. y_i are Gaussian distributed since the received signal power in shadowing is assumed to be log-normally distributed [89]. The observations $y_i, i = 1, \ldots, L$, may be correlated depending on the location of the CR users. The collection of these observations is the Gaussian distributed vector $\mathbf{Y} = \{y_i\}_1^L$ under the null hypothesis H_0, which indicates the absence of the PU transmit signal, and

the alternative hypothesis H_1, which indicates the presence of the transmit signal, as follows [89]:

$$\mathbf{Y} \sim \begin{cases} \mathcal{N}(\mathbf{0}, \sigma_0^2 \boldsymbol{I}), & H_0 \\ \mathcal{N}(\boldsymbol{\mu_1}, \sigma_1^2 \boldsymbol{\Sigma}), & H_1 \end{cases}, \tag{5.1}$$

where $\mathbf{0}$ is the zero vector, $\boldsymbol{\mu_1}$ is the mean SNR that depends on the path loss from the location of the PU, σ_0^2 is the Gaussian noise variance under H_0, σ_1^2 is the variance of noise in correlated shadowing under H_1, $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{\Sigma}$ is the normalized covariance with elements ρ_{ij}. We assume that the correlation follows the exponential correlation model [35]. In this model, the correlation coefficients can be expressed as

$$\rho_{ij} = e^{-d_{ij}/D_c} = e^{-a \cdot d_{ij}}, \tag{5.2}$$

where d_{ij} is the distance between CR users i and j, D_c is the de-correlation distance, and $a = 1/D_c$ is the exponential decaying coefficient set to 0.1204 and 0.002 for urban and suburban settings, respectively [32]. Thus, two CR users are correlated if the distance between them is smaller than D_c, and uncorrelated otherwise.

Depending on the distance between the PU and the CR user, and the degree of fading, the SNR observed at CR users may vary significantly. Due to these variations, CR users may take different number of observations to satisfy detection requirements and report their local decisions on the CCC *asynchronously* that causes different reporting delays. We assume that a narrowband CCC is shared by the FC for message broadcast and by cooperating CR users for reporting local sensing data. The time-varying wireless channel between each CR user and the FC, known as the reporting channel, is susceptible to independent Rayleigh fading, which is modeled by the finite-state Markov channel (FSMC) model [95]. Since the received SNR γ_b varies with time and ranges from 0 to infinity, the entire SNR range is divided into J regions in which the jth region is defined as $\Gamma_j = [A_j, A_{j+1}) = \{\gamma_b : A_j \leq \gamma_b < A_{j+1}\}$ where $\{A_j\}$ are region boundaries with $A_0 = 0$ and $A_J = \infty$. For transmitting binary local decisions,

we assume BPSK modulation at reporting CR user and the coherent demodulation at the FC. In this case, the error probability can be expressed in terms of the received SNR as $P_e(\gamma_b) = Q(\sqrt{2\gamma_b})$ where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-u^2/2} \mathrm{d}u$ is the tail probability of standard normal distribution. The reported local decision in each channel state x_j follows a binary symmetric channel (BSC) where the local decisions are received in errors at the FC with crossover probability ε_j.

Based on the proposed RLCS model and algorithm discussed in Section 5.3, the FC selects and combines these local results, and makes a cooperative decision on the presence of the PU. In general, the data fusion of selected $K \leq L$ local sensing results at FC is given by

$$\sum_{i=1}^{K} w_i f(y_i) = \sum_{i=1}^{K} w_i u_i \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \lambda_0 \tag{5.3}$$

where $f(\cdot)$ is the local decision process, w_i is the weighting factor for local sensing data $u_i \in \{0, 1\}$ from cooperating CR user i and λ_0 is the cooperative decision threshold at the FC. For hard combinations with the majority rule, $w_i = 1, \forall i$ and $\lambda_0 = \lceil K/2 \rceil$. The majority rule is chosen over *AND* and *OR* rules for the balance of false alarms and miss detection. The cooperative decision $u_0 \in \{0, 1\}$ is then broadcast to all neighbors. This cooperative sensing process is periodically repeated for infinite iterations, called *episodes*.

5.3 Reinforcement Learning-Based Cooperative Sensing

In this section, we present the proposed RLCS model and algorithm for cooperative sensing. We formulate the problem as a cooperative sensing decision process (CSDP) and discuss the process of RLCS algorithm for improving cooperative gain in cooperative sensing.

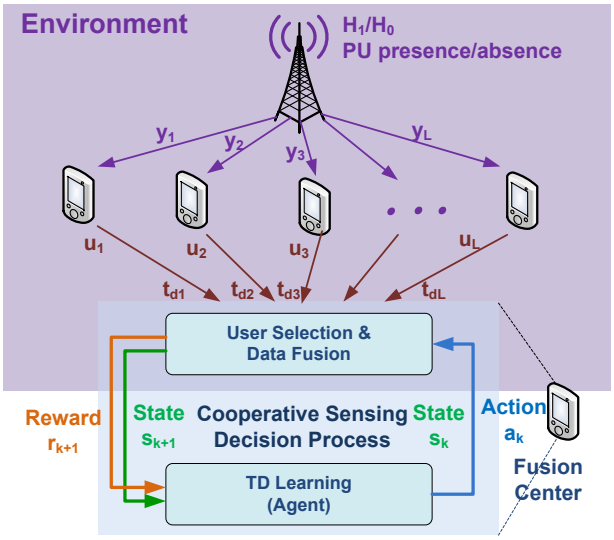


Figure 5.2: Model of Cooperative Sensing with Reinforcement Learning.

5.3.1 Cooperative Sensing Decision Process

In RLCS, the interactions between the FC and cooperative CR users are modeled as a CSDP. CSDP is a decision process with non-Markovian rewards for FC's sequential decisions on selecting cooperating neighbors. Figure 5.2 illustrates the RLCS model with the inherent CSDP and the environment with which the agent inside the FC interacts. In the figure, the FC interacts with L cooperating neighboring nodes that observe PU signals in the environment and obtain Gaussian distributed and possibly correlated observations $\mathbf{Y} = \{y_i\}_1^L$ as in (5.1). In each state s_k, where k is the time step or stage index, the FC selects neighbor i by choosing action $a_k = i$ and receives local decision u_i determined from observed y_i with reporting delay t_{d_i} as reward r_{k+1} along with state change to s_{k+1}. By exploring the unknown states and accumulating the knowledge of receiving rewards from known states, the FC learns the sequence of optimal decision rules, the optimal policy that gives rise to the maximum reward.

The CSDP is represented by a quadruple $\langle \mathcal{S}, \mathcal{A}, f_p, f_R \rangle$ in which $\mathcal{S} = \{0, 1, \ldots, L\}$ is a finite set of all states, $\mathcal{A} = \cup_{j \in \mathcal{S}} \mathcal{A}_j = \{0, 1, \ldots, L\}$ is a finite set of actions, where $\mathcal{A}_j \subseteq \mathcal{A}$ is the set of actions available in state j, f_p is the state transition probability

function, and f_R is the reward function. Each component is described as follows:

States: A state of the CSDP is the status of user selection and reporting in the environment that includes cooperating CR users and their observations of the PU signal. In each episode n, the states of the environment s_k, which take on values from $\mathcal{S} = \{0, 1, \ldots, L\}$, are defined as

$$s_k^n = i \cdot \mathcal{I}\{a_{k-1}^n = i \in \mathcal{A}_j, s_{k-1}^n = j \in \mathcal{S}, i \neq j, k \neq 0\}, \tag{5.4}$$

where $\mathcal{I}\{x\}$ is the indicator function and equals one if x is true and zero otherwise, and a_{k-1}^n is the action selected in s_{k-1}^n. The process starts with the FC state (CR user 0 state) $s_0^n = 0$ when the FC initiates the process of cooperative sensing at time t_0^n. In each state $s_k^n = j \in \mathcal{S}$, the FC requests local decision u_i from CR user i and awaits a response from the environment by choosing action $a_k^n = i \in \mathcal{A}_j$, or terminates the cooperative sensing by choosing action $a_k^n = 0$ to return to the FC state. The state changes from $s_k^n = j$ to $s_{k+1}^n = i$ when the FC obtains reported u_i and the corresponding reward r_{k+1}^n as the response at time t_{k+1}^n. The FC state is both the *start* state and the *terminal* state in each episode.

Actions: An action is the FC's decision on selecting a CR user (including the FC itself) for reporting in a state. Let $h_k^n = (s_0^n, a_0^n, \ldots, s_{k-1}^n, a_{k-1}^n, s_k^n)$ be the history of state-action sequence from s_0^n to s_k^n in episode n. The decision rule μ_k^n is the function mapping h_k^n into a probability distribution $\Delta_{\mu_k^n}(\mathcal{A}_{s_k^n})$ on the set of actions $\mathcal{A}_{s_k^n}$ in state s_k^n of episode n. Let also $\mathcal{D}_{k-1}^n$ be the set of selected CR users from s_0^n to s_{k-1}^n of episode n given by $\mathcal{D}_{k-1}^n = \{a_0^n, \ldots, a_{k-1}^n\}$ and $\mathcal{D}_{-1}^n = \emptyset$. Thus, the actions of the FC a_k^n in state s_k^n, which take on values from $\mathcal{A}_{s_k^n} = \mathcal{A} \setminus \{\{s_k^n\} \cup \mathcal{D}_{k-1}^n\}$, are defined as:

$$a_k^n = \mu_k^n(h_k^n) \in \mathcal{A}_{s_k^n}, \quad \text{w.p. } p(s_k^n, \mu_k^n(h_k^n)) \tag{5.5}$$

where $p(s_k^n, \mu_k^n(h_k^n))$, defined in (5.9), is the probability of selecting a_k^n in s_k^n according to $\Delta_{\mu_k^n}(\mathcal{A}_{s_k^n})$. In each state of episode n, the FC selects CR user i with probability

$p(s_k^n, a_k^n = i)$ for reporting in s_k^n. Specifically, the FC requests cooperating CR user i to report local decision u_i by sending $a_k^n = i$, or informs all cooperating CR users the cooperative decision u_0 by sending $a_k^n = 0$ along with u_0. In the latter, action $a_k^n = 0$ also terminates one round of cooperative sensing. Nevertheless, how to choose the actions depends on the action selection strategy defined in Section 5.3.2.

Transition Probability Function: The transition probability function, $f_p : \mathcal{H} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ where $\mathcal{H} = \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \cdots \times \mathcal{S}$, maps the state-action-state transitions to a probability of changing from the current state to the next state by choosing the action. The transition probability from state $s_k^n = j$ to $s_{k+1}^n = i$ by choosing action $a_k^n = i$ in s_k^n is denoted by $p_{ji} = P(i|j, \mu_k^n(h_k^n))$ and is generally not known a priori. Since a chosen action implies the transition to a particular state in our model, the probability of choosing an action $a_k^n = i$ in state $s_k^n = j$ can be considered as the state transition probability from $s_k^n = j$ to $s_{k+1}^n = i$. As a result, the FC gradually learns the state transitions from the action selection probabilities, even though the transition probabilities are not required by TD learning algorithms.

Reward Function: The reward function, $f_R : \mathcal{H} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, maps the state-action-state transitions to a real-valued reward. The reward is used by the FC to evaluate action selections for choosing uncorrelated CR users with small reporting delay for cooperative sensing. The FC receives a reward r_{k+1}^n upon the arrival of the local sensing data u_i from CR user i with reporting delay $t_{d_i}^n$ as the result of action $a_k^n = i$ in state $s_k^n = j$. The reward r_{k+1}^n corresponding to the action a_k^n in state s_k^n of episode n is given by:

$$r_{k+1}^n(s_k^n, \mu_k^n(h_k^n)) = r_{\rho_{k+1}}^n \mathcal{I}\{C_{\rho_{k+1}}^n \neq 0\} + r_{d_{k+1}}^n \mathcal{I}\{C_{\rho_{k+1}}^n = 0\}, \quad k = 0, \ldots, K^n - 1, \tag{5.6}$$

where $r_{\rho_{k+1}}^n = -C_{\rho_{k+1}}^n$ and $r_{d_{k+1}}^n = 1 - C_{d_{k+1}}^n$ are the rewards attributed to correlation cost $C_{\rho_{k+1}}^n$ in (5.7) and delay cost $C_{d_{k+1}}^n$ in (5.8), respectively, and $K^n \leq L$ is the number of selected cooperating CR users in episode n. Note that $r_{k+1}^n = 0$ for $K^n \leq k \leq L$. (5.6) states that r_{k+1}^n is determined by the delay cost if the selected CR

user is uncorrelated with previously selected CR users, and by the correlation cost if the selected CR user incurs correlation. r_{k+1}^n is positive only when selected CR users are uncorrelated and their cumulative reporting delay is within the delay constraint.

The correlation between CR users' observations in correlated shadowing is captured by the covariance matrix $\mathbf{\Sigma}$ in (5.1). The elements of $\mathbf{\Sigma}$, correlation coefficients ρ_{ij}, are estimated by using location information and (5.2). These correlation coefficients affect the correlation cost (5.7) and the reward (5.6) obtained in each state. Given a different CR user j, $j \neq i$ selected in state $s_l, l = 0, \ldots, k-1$ and $\mathbf{\Sigma} = \{\rho_{ij}\}$, correlation cost $C_{\rho_{k+1}}$ is given by:

$$C_{\rho_{k+1}}^n = \left[\frac{1}{k}\sum_{\ell=0}^{k-1} |\rho_{ij}(s_\ell, a_\ell = j)|\right] \mathcal{I}\{k > 0\}, \quad j \neq i. \tag{5.7}$$

Thus, the correlation cost is simply the average of correlation coefficients between the newly selected CR user i and each selected CR user in previous k states.

The delay cost $C_{d_{k+1}}^n$, on the other hand, is attributed to reporting delays. The reporting delay of CR user i in s_k^n, $t_{d_i}^n = t_d(s_k^n, a_k^n = i)$, is the interval between the time of the FC requesting CR user i's cooperation with the action $a_k^n = i$ and the arrival time of the local sensing data u_i at the FC. Thus, the delay cost $C_{d_{k+1}}^n$ is given by:

$$C_{d_{k+1}}^n = \frac{\sum_{\ell=0}^{k-1} t_d(s_\ell^n, a_\ell^n = j) + t_d(s_k^n, a_k^n = i)}{T_{lim}}, \quad i \in A_{s_k}, j \in A_{s_\ell}, i \neq j \tag{5.8}$$

where $T_{lim} = \min\{T_{c_{max}}, T_{d_{avg}}\}$ is the total reporting delay constraint, $T_{c_{max}}$ is the maximum allowed cooperative sensing time, and $T_{d_{avg}} = \sum_{j=1}^{L} \bar{t}_{d,j}$ is the total average reporting delay of all CR users. It is simply the cumulative reporting time up to the start of next state s_{k+1}^n normalized by the factor of maximum cooperative sensing time or total average reporting delay, whichever is smaller. This means that the reward attributed to the delay cost $r_{d_{k+1}}^n$ is lower for CR users to be selected in the later stage than the earlier stage, which enforces the CR user with large average reporting

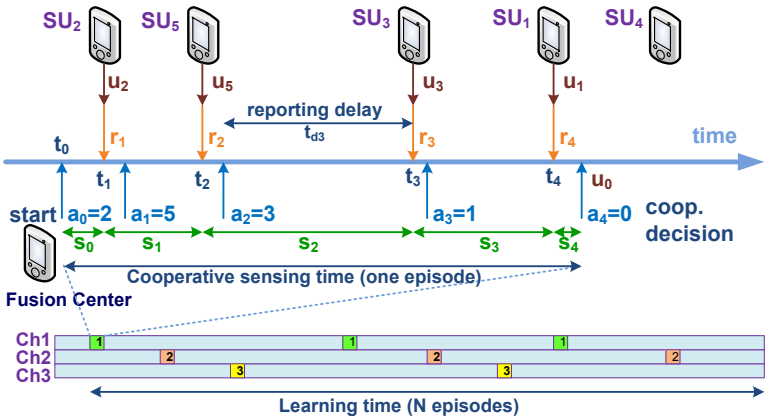


Figure 5.3: Example of One RL-Based Cooperative Sensing Episode with the CSDP.

delay to be less attractive for participation, especially in the later stage when the cooperative decision needs to be determined within the limit $T_{c_{max}}$.

From (5.7), (5.8) and the definition of r_{k+1}^n in (5.6), we know that negative rewards are obtained when the selected CR users are correlated or their cumulative reporting delays exceed the maximum tolerable cooperative sensing time. Such selections are learned and will be depreciated from future selections. Positive rewards are possible only when all selected CR users are uncorrelated. Thus, large positive rewards are more likely attributed to selecting more uncorrelated CR users with small reporting delays within the time constraint $T_{c_{max}}$.

Figure 5.3 gives an example of the CSDP. The FC initially starts in state s_0. After choosing the first action $a_0 = 2$, it waits for the first reward r_1. SU_2 responds to the request with its decision u_2. Upon the receipt of reward r_1 calculated from u_2, the state changes to the next state $s_1 = 2$. This cycle continues from state s_1 to state s_4. In s_4, the agent chooses the action $a_4 = 0$ to terminate the cooperative sensing process. Note that SU_4 is not selected in this case.

5.3.2 RL-Based Cooperative Sensing Algorithm

Based on the CSDP model, the RLCS algorithm learns the environment by iteratively choosing actions, receiving rewards, and evaluating action selections with the

objectives of maximizing received rewards and improving cooperative gain. In the following, we discuss action selection strategy, expected cumulated reward for optimal policy, state-action value updates for action evaluation, and user selection for reliable cooperation.

5.3.2.1 Action Selection Strategy

The action selection strategy affects how the FC interacts with the environment. In RLCS, the softmax approach based on Boltzmann distribution is utilized for action selections. In this action strategy, the probability of selecting action $a_k^n = i$ in state s_k^n is given by:

$$p(s_k^n, a_k^n = i) = \frac{e^{Q(s_k^n, a_k^n = i)/\tau^n}}{\sum_{j=1}^{|\mathcal{A}_{s_k^n}|} e^{Q(s_k^n, a_k^n = j)/\tau^n}}, \quad i \in \mathcal{A}_{s_k^n} \tag{5.9}$$

where $Q(s_k^n, a_k^n)$ is the state-action value (*Q-value*) function that evaluates the quality of choosing action a_k^n in state s_k^n, and τ^n is an episode-varying parameter called *temperature* that controls the degree of exploration versus exploitation. For large values of τ^n, all actions are equally probable. In this case, the FC explores the opportunities of more uncorrelated cooperating CR users to achieve potentially higher detection probability in the future with large τ^n. For small τ^n, on the other hand, the action with maximum $Q(s, a)$ is favored. Hence, the agent exploits the current knowledge of best selections of cooperating CR users to achieve the potentially highest detection probability with small τ^n. As a result, τ^n remains a large value for exploration in a highly dynamic environment while τ^n is decreased to a small value for exploitation in a static environment where the convergence can be assured [80]. To achieve the convergence in a certain number of episodes, we use a linear function to decrease the value of τ^n over episodes as follows:

$$\tau^n = -(\tau^0 - \tau^N) \cdot n/N + \tau^0, \quad n \leq N \tag{5.10}$$

where N is the number of episodes to reach the convergence, τ^0 and τ^N are the initial and the last value, respectively, of the temperature in N episodes. Note that $\tau^n \neq 0$,

$\forall n$ and $\tau^n = \tau^N \approx 0$ for $n \geq N$ until any environmental change.

5.3.2.2 Expected Cumulative Reward

The expected cumulative reward R^n of episode n is defined as

$$R^n = E[Y^n] = E\Big[\sum_{k=0}^{L} r_{k+1}^n(s_k^n, \mu_k^n(h_k^n))\Big], \tag{5.11}$$

where $Y^n = \sum_{k=0}^{L} r_{k+1}^n$ is the cumulative reward of episode n. If there are K^n CR users selected in episode n, $Y^n = \sum_{k=0}^{K^n} r_{k+1}^n$ and $r_{k+1}^n = 0$ for $k = K^n, \ldots, L$. The objective of TD learning in RLCS is to find the optimal policy $\pi^* = \{\mu_0^*, \ldots, \mu_{K^*}^*\}$, where $K^* \leq L$ is the optimal number of selected cooperating CR users, to achieve the maximum cumulative reward R_{π^*}, which leads to higher detection performance in cooperative sensing.

5.3.2.3 State-Action Value Updates

To evaluate the quality of action selections, a table known as *Q-table* of size $|\mathcal{S}| \times |\mathcal{A}|$ is used to store the Q-values for all state-action pairs. In each state s_k^n with the selection of action a_k^n, the Q-value $Q(s_k^n, a_k^n)$ for the state-action pair (s_k^n, a_k^n) needs to be updated according to the received reward r_{k+1}^n and future state-action value estimates $Q(s_{k+1}^n, a_{k+1}^n)$. The general form of the Q-value update in TD learning can be expressed as

$$Q_k^n \leftarrow (1 - \alpha_k^n)Q_k^n + \alpha_k^n\big[r_{k+1}^n + \gamma f(Q_{k+1}^n)\big], \tag{5.12}$$

where $Q_k^n = Q(s_k^n, a_k^n)$, α_k^n is the learning rate, γ is the discount factor for future state-action value estimates Q_{k+1}^n, and $f(Q_{k+1}^n)$ is the function of future estimates Q_{k+1}^n, and the function depends on the TD learning algorithms used. For example, $f(Q_{k+1}^n) = \max_{a_{k+1}^n} Q_{k+1}^n$ and $f(Q_{k+1}^n) = Q_{k+1}^n$ are future estimate functions for Q-learning [97] and Sarsa [85], respectively. The discount factor γ determines the weight of the future Q-value estimates compared to the current Q-value for the (s_k^n, a_k^n) pair. For faster convergence, the learning rate α_k^n is decreased as the (s_k^n, a_k^n) pair is explored

more often. However, α_k^n should remain constant or sufficiently large to take into account the latest changes in the highly dynamic environment that may be caused by, for example, the movement of CR users.

5.3.2.4 User Selection for Reliable Cooperation

The selection of cooperating users ensures that reliable users can be constantly selected to participate in cooperative sensing and contribute correct local decisions to improve detection performance while the unreliable ones are excluded. Let $p(\mathbf{u}_i)$ be the distribution of local decisions reported from CR user i and $p(\mathbf{u}_0)$ be the distribution of cooperative decisions at the FC. These are CR users' and FC's estimates of PU activity P_{on} and P_{off}. After receiving CR user i's report u_i and having the cooperative decision u_0, the FC will update $p(\mathbf{u}_i)$ and $p(\mathbf{u}_0)$, respectively. Based on the assumption that cooperative decisions are statistically more accurate than local decisions, we use the *Kullback-Leibler (KL) distance*, $D(p(\mathbf{u}_0)\|p(\mathbf{u}_i))$, to measure how far the distribution of local decisions $p(\mathbf{u}_i)$ diverges from the distribution of cooperative decisions $p(\mathbf{u}_0)$ at the FC. Since $D(p(\mathbf{u}_0)\|p(\mathbf{u}_i)) \geq 0$ and is zero when $p(\mathbf{u}_0) = p(\mathbf{u}_i))$, larger DL distance indicates that the degree of the divergence is higher and implies that CR user i is more unreliable. To determine the user reliability, we compare the KL distance with a threshold δ_{DL} and claim that CR user i is considered unreliable for cooperative sensing if

$$D(p(\mathbf{u}_0)\|p(\mathbf{u}_i)) = \sum_{u_i \in \{0,1\}} p(u_i) \log \frac{p(u_i)}{p(u_f)} > \delta_{DL} \tag{5.13}$$

and reliable otherwise. The threshold δ_{DL} is set to $\mu_\delta + c\sigma_\delta$, where μ_δ and σ_δ are the mean and the standard deviation, respectively, of the DL distances of all reliable cooperating CR users, and c is a constant. Let $\mathcal{U} \subseteq \mathcal{C}$ be the set of uncorrelated CR users selected from the first user selection step. The set of CR users selected for data fusion $\mathcal{D}$ is the subset of $\mathcal{U}$ given by

$$\mathcal{D} = \mathcal{U} \cap \{\cup_i \{i \mid D(p(\mathbf{u}_i)\|p(\mathbf{u}_f)) \leq \delta_{DL}, \forall i\}\}. \tag{5.14}$$

Thus, a previously unreliable CR user will be included in $\mathcal{D}$ for cooperation when it becomes reliable and satisfies the condition in (5.14).

5.3.2.5 RLCS Algorithm

The RLCS algorithm is listed in Algorithm 5.1. The algorithm takes the array of all state-action values ($Q(|\mathcal{S}|, |\mathcal{A}|)$) as the input and initializes the array entries to zero. The output is the optimal solution: optimal number of cooperating CR users ($K^* = |\mathcal{P}|^*$), optimal reporting sequence (π^*), and total reporting delay (T_d^*). In the algorithm, RLCS is performed repeatedly (line 3 to 19) unless the optimal stopping criterion is met. In each episode, there are $K^n \leq L$ CR users selected for sensing and reporting based on the action strategy. In each state of an episode (line 6-14), the FC sends the request to the selected CR user, receives the local decision, calculates the reward, and updates the Q-value using (5.12). At the end of the episode, the FC terminates the episode with $a_k^n = 0$, determines the reliable set of CR users and the cooperative sensing decision, and broadcasts it to all neighbors (line 16-18).

5.4 Performance Analysis

In this section, we analyze the performance of the proposed RLCS scheme by first showing the optimal solution of RLCS, proving its convergence, and evaluating the rate of convergence. We then provide the optimal stopping alternative for performance improvement in expected cumulative rewards. Lastly, we discuss the impact of control channel fading on reporting errors and detection performance.

5.4.1 Optimal Solution of RLCS Algorithm

The RLCS algorithm is capable of learning the changes in a dynamic environment to reach the optimal solution. The optimal solution of RLCS is the optimal set of spatially uncorrelated CR users, $\mathcal{P}^* \subseteq \mathcal{C}$, selected for cooperation in sequence by optimal policy $\pi^* = (\mu_0, \ldots, \mu_{K^*})$ that achieves maximum cumulative reward R_{π^*},

Algorithm 5.1 : RL-based Cooperative Sensing (RLCS)

Input: $Q(|\mathcal{S}|, |\mathcal{A}|), L, N$
Output: $\pi^* \leftarrow \pi$, $\mathcal{P}^* \leftarrow \mathcal{D}^n, K^* \leftarrow |\mathcal{P}^*|, T_d^* \leftarrow T_d$
repeat
  Initialize: $\mathcal{D}^n = \mathcal{U}^n = \emptyset, K^n = 0, T_d = 0$
  for $k \leftarrow 0$ to $(L+1)$ **do**
    $a_k^n \leftarrow$ *ActionStrategy*$(\mu_k^n, s_k^n, Q, \tau^n)$
    if $a_k^n \neq 0$ **then**
      SendReqToNeighbor$(CRuser_i; a_k^n = i)$
      $r_{k+1}^n \leftarrow$ *WaitForReward*$(u_i; t_d^n(s_k, a_k = i))$
      $Q_k^n \leftarrow (1 - \alpha_k^n) Q_k^n + \alpha_k^n \left[r_{k+1}^n + \gamma f(Q_{k+1}^n) \right]$
      Update$(\pi, \alpha_k^n, \bar{t}_{d_i}, T_d, K^n, Y^n, \mathcal{U}^n)$
    else
      break
    end if
  end for
  $\mathcal{D}^n = \mathcal{U}^n \cap \{\cup_i \{i \mid D(p(\mathbf{u}_i) \| p(\mathbf{u}_f)) \leq \delta_{DL}, \forall i\}\}$
  $u_0 \leftarrow$ *CoopSensingDecision*$(u_i, k, L, \forall i \in \mathcal{D}^n)$
  BroadcastCoopDecision$(u_0; a_k = 0)$
until $Y^n \geq \tilde{Y}^*$

where $K^* = |\mathcal{P}^*|$ is the optimal number of selected CR users. In the following, we present the necessary conditions of achieving the optimal solution in a static environment, where CR user locations and their reporting delays are known, as two lemmas followed by the corresponding theorem.

Lemma 5.1 (Optimal Number of Selected Users)**.** *Given the set of L cooperating CR users, $\mathcal{C}$, with their locations, the optimal number of selected CR users, $K \leq L$, is the maximum number of spatially uncorrelated CR users that maximize $R_{\rho_L} = \sum_{k=0}^{L-1} r_{\rho_{k+1}}$ with maximum value $R_{\rho_K} = 0$.*

Proof. Let $\mathcal{P}_k$ be the set of selected CR users from s_0 to s_k and CR user j, $j \in \{\mathcal{C} \setminus \mathcal{P}_{k-1}\}$, be the CR user selected in s_k. From (5.6) and (5.7), $r_{\rho_{k+1}} = -C_{\rho_{k+1}}$ and $C_{\rho_{k+1}} \geq 0$. The maximum value of $r_{\rho_{k+1}}$ is 0 and can be obtained if and only if $C_{\rho_{k+1}} = 0$. By (5.7), $C_{\rho_{k+1}} = \frac{1}{k} \sum_{\ell=0}^{k-1} |\rho_{ij}(s_\ell, a_\ell = i)| = 0$, $\forall i \in \mathcal{P}_{k-1}$. All selected CR users in $\mathcal{P}_k$ must be spatially uncorrelated to maximize $r_{\rho_{k+1}}$. If CR user j is spatially correlated to any CR user in $\mathcal{P}_{k-1}$, $r_{\rho_{k+1}} < 0$. Thus, $r_{\rho_{k+1}} = 0$ for $k = 0, \ldots, K-1$ for

selecting up to maximum $K \leq L$ spatially uncorrelated CR users in $\mathcal{C}$ to maximize R_{ρ_L} with maximum value $R_{\rho_K} = 0$ as $r_{\rho_{k+1}} < 0$, $k = K, \ldots, L-1$, such that $R_{\rho_k} < R_{\rho_K}$, $k = K+1, \ldots, L$. □

Lemma 5.2 (Optimal User Selection Sequence)**.** *Given the set of K selected CR users, $\mathcal{P}$, with their reporting delays t_{d_i}, $i \in \mathcal{P}$, there exists an optimal user selection sequence $\pi^*_{K^*} = (a^*_0, \ldots, a^*_{K^*-1})$ that maximizes $R_{d_K} = \sum_{k=0}^{K-1} r_{d_{k+1}}$,*
where $a^*_k = \arg\min_{a_k \in \mathcal{A}_{s_k}} t_d(s_k, a_k)$ and $K^* \leq K$ is the maximum number of selected CR users that satisfy total reporting delay constraint $T_{d_{K^*}} = \sum_{k=0}^{K^*-1} t_d(s_k, a^*_k) \leq T_{lim}$.

Proof. Let $d_{k,i} = t_{d_i}(s_k, a_k)$, $i \in \mathcal{A}_{s_k}$, be the reporting delay of CR uesr i selected in s_k. From (5.6) and (5.8),

$$r_{d_{k+1}} = 1 - C_{d_{k+1}} = 1 - \frac{\sum_{\ell=0}^{k} d_{\ell,i}}{T_{lim}} = \left(1 - \frac{\sum_{\ell=0}^{k-1} d_{\ell,i}}{T_{lim}}\right) - \frac{d_{k,i}}{T_{lim}} = r_{d_k} - \frac{d_{k,i}}{T_{lim}}. \quad (5.15)$$

If $T_{d_{k+1}} = \sum_{\ell=0}^{k} d_{\ell,i} \leq T_{lim}$, we have $0 < C_{d_{k+1}} \leq 1$ and $r_{d_{k+1}} \geq 0$. Since $d_{k,i}, T_{lim} > 0$ and T_{lim} is constant, maximizing $r_{d_{k+1}}$ is equivalent to minimizing $d_{k,i}$ given r_{d_k} obtained in the previous state. As a result, the optimal user selection that maximizes $r_{d_{k+1}}$ in state s_k is $a^*_k = \arg\min_{i \in \mathcal{A}_{s_k}} d_{k,i}$, $k = 0, \ldots, K^*-1$. Since K^* is the maximum number of CR users that satisfy $T_{d_{K^*}} \leq T_{lim}$, we have $T_{d_{k+1}} > T_{lim}$ and $r_{d_{k+1}} < 0$ for $k = K^*, \ldots, K-1$, due to $C_{d_{k+1}} > 1$ from (5.8). Thus, there exists an optimal user selection sequence $\pi^*_{K^*} = (a^*_0, \ldots, a^*_{K^*-1})$, where $a^*_k = \arg\min_{i \in \mathcal{A}_{s_k}} d_{k,i}$, that maximizes R_{d_K} and achieves maximum value $R_{d_{K^*}} = \sum_{k=0}^{K^*-1} r_{d_{k+1}}$. □

Theorem 5.1 (Optimal Solution of RLCS Algorithm)**.** *The optimal solution of RLCS is a 3-tuple $\langle K^*, \mathcal{P}^*, T^*_d \rangle$ obtained by the optimal policy $\pi^* = \{\mu^*_0, \ldots, \mu^*_{K^*-1}\}$ that achieves the maximum cumulative reward given by*

$$R_{\pi^*} = K^* - \frac{\sum_{k=0}^{K^*-1} (K^* - k) \cdot t_d(s_k, \mu^*_k)}{T_{lim}} \quad (5.16)$$

and total reporting delay $T^*_d = \sum_{k=0}^{K^*-1} t_d(s_k, \mu^*_k) < T_{lim}$,
where $\mu^*_k = \arg\min_{a_k \in \mathcal{A}_{s_k}} t_d(s_k, a_k)$.

Proof. In (5.11), r_{k+1} received in s_k can be negative if $C_{\rho_{k+1}} > 0$ in (5.7) or $C_{d_{k+1}} > 1$ in (5.8). That is, the CR user selected in s_k either causes spatial correlation with any previously selected CR users or incurs reporting delay that does not satisfy the total reporting delay constraint T_{lim}. To avoid negative rewards, $C_{\rho_{k+1}} = 0$ and $0 < C_{d_{k+1}} \leq 1$ are required. Using Lemma 5.1, we obtain maximum K spatially uncorrelated CR users to ensure $r_{\rho_{k+1}} = C_{\rho_{k+1}} = 0$. By plugging (5.6) into (5.11) and setting $C_{\rho_{k+1}} = 0$, R_π is reduced to $R_\pi = \sum_{k=0}^{K-1} r_{d_{k+1}} = R_{d_K}$. Using Lemma 5.2, we obtain optimal user selection sequence $\pi^*_{K^*}$ to maximize R_π and ensure $0 < C_{d_{k+1}} \leq 1$ with maximum K^* selected CR users that satisfy $T_{d_{K^*}} < T_{lim}$. Since the decision rules μ_k are deterministic, $\mu^*_k = a^*_k = \arg\min_{a_k \in \mathcal{A}_{s_k}} t_d(s_k, a_k)$ in $\pi^* = \pi^*_{K^*}$ with $T^*_d = T_{d_{K^*}}$ to achieve

$$R_{\pi^*} = R^*_{d_{K^*}} = \sum_{k=0}^{K^*-1} r^*_{d_{k+1}} = K^* - \sum_{k=0}^{K^*-1} C^*_{d_{k+1}} = K^* - \frac{1}{T_{lim}} \sum_{k=0}^{K^*-1} \sum_{\ell=0}^{k} t_d(s_\ell, \mu^*_\ell). \quad (5.17)$$

After some algebraic manipulation, (5.16) follows. □

5.4.2 Convergence of RLCS Algorithm

The optimal solution is achieved when the RLCS algorithm converges with sufficient exploration of state-action pairs. In the case of insufficient exploration, a suboptimal solution may be obtained upon convergence. To prove the convergence, we first show that the sequence of expected cumulative rewards $\{R^n\}$ is a submartingale in Lemma 5.3, and the result follows the Martingale convergence theorem given in Lemma 5.4.

Lemma 5.3 (Sequence of Expected Cumulative Rewards)**.** *The sequence of expected cumulative rewards R^n, $n = 1, 2, \ldots$ is a submartingale that satisfies $E|R^n| < \infty$ and $E[R^{n+1}|R^i] \geq R^n, \forall i \leq n$.*

Proof. From (5.11), we can easily show that $E|R^n| < \infty, n = 1, 2, \ldots$ since L and r^n_{k+1} are finite. Next, we show $E[R^{n+1}|R^i] \geq R^n$, $\forall i \leq n$. Let Π be the set of all

policies.

$$E[R^{n+1}|R^i] = \sum_{\pi\in\Pi} R_\pi^{n+1} p(R_\pi^{n+1}|R^i) \tag{5.18}$$

$$= \sum_{\pi\in\Pi}\sum_{k=0}^{L} E\Big[r_{k+1}^{n+1}(s_k^{n+1},\mu_k^{n+1})\Big] p(R_\pi^{n+1}|R^i) \tag{5.19}$$

$$= \sum_{\pi\in\Pi}\sum_{k=0}^{L}\sum_{s_k^{n+1},a_k^{n+1}} r_{k+1}^{n+1}(s_k^{n+1},\mu_k^{n+1}) p_r(s_k^{n+1},\mu_k^{n+1}) p(R_\pi^{n+1}|R^i) \tag{5.20}$$

$$= \sum_{\pi\in\Pi}\sum_{k=0}^{L}\sum_{s_k^{n+1},a_k^{n+1}} r_{k+1}^{n+1}(s_k^{n+1},a_k^{n+1}) p_\pi(s_{k+1}^{n+1}|s_k^{n+1},a_k^{n+1}) \tag{5.21}$$

where R_π^{n+1} in (5.18) is the expected cumulative reward obtained by π, p_r in (5.20) is reward distribution for π, and p_π in (5.21) is the transition probability for π in episode $n+1$ given R_i, equivalently, all Q-value updates with r_{k+1}^i, $k=0,\ldots,L$, $i\le n$, in previous n episodes. Since, as indicated in Section 5.3, the state transition probability is the action selection probability and let $s_k^{n+1}=x$ and $a_k^{n+1}=s_{k+1}^{n+1}=y$, from (5.9), we have

$$p_{xy}^{n+1} = \frac{e^{Q^{n+1}(s_k=x,a_k=y)/\tau^{n+1}}}{\sum_{j=1}^{|\mathcal{A}_x|} e^{Q^{n+1}(s_k=x,a_k=j)/\tau^{n+1}}}, \quad y\in\mathcal{A}_x. \tag{5.22}$$

We now compare $r_{k+1}^{n+1}(x,y)p_{xy}^{n+1}$ with $r_{k+1}^{n}(x,y)p_{xy}^{n}$ for a policy π. For simplicity, we first assume no future estimates ($\gamma=0$). Since the Q-value for each (x,y) is updated no more than once in each episode, Q^{n+1} before the update in episode $n+1$ equals Q^n after the update (5.12) in episode n. Hence, (5.12) is simplified to

$$Q_k^{n+1} = (1-\alpha)Q_k^n + \alpha r_{k+1}^n = Q_k^n + \alpha(r_{k+1}^n - Q_k^n). \tag{5.23}$$

If the delays and locations of CR users are fixed, the reward of the same state-action pair (x,y) in a policy π is the same for different episodes: $r_{k+1}^{n+1}(x,y) = r_{k+1}^n(x,y)$. Moreover, p_{xy}^{n+1} is the function of $\frac{Q^{n+1}}{\tau^{n+1}}$ and we know from (5.10) that $\tau^{n+1}<\tau^n$. As a result, depending on how Q-values change from episode n to $n+1$, we have the following six cases:

(i) $Q^{n+1} \geq Q^n \geq 0$: In this case, $p_{xy}^{n+1} > p_{xy}^n$ because $\frac{Q^{n+1}}{\tau^{n+1}} > \frac{Q^n}{\tau^n}$. From (5.23), $Q_k^{n+1} - Q_k^n = \alpha(r_{k+1}^n - Q_k^n) \geq 0$. We have $r_{k+1}^{n+1} = r_{k+1}^n \geq Q_k^n \geq 0$. Thus, $r_{k+1}^{n+1} p_{xy}^{n+1} \geq r_{k+1}^n p_{xy}^n$.

(ii) $Q^{n+1} \geq 0 \geq Q^n$: As in case (i), $p_{xy}^{n+1} > p_{xy}^n$. Using (5.23) and $Q_k^n \leq 0$, we obtain $r_{k+1}^{n+1} = r_{k+1}^n \geq (1 - \frac{1}{\alpha})Q_k^n \geq 0$. Thus, $r_{k+1}^{n+1} p_{xy}^{n+1} \geq r_{k+1}^n p_{xy}^n$.

(iii) $0 \geq Q^{n+1} \geq Q^n$: Similarly, $p_{xy}^{n+1} > p_{xy}^n$. In this case, $r_{k+1}^n \in [Q_k^n, (1 - \frac{1}{\alpha})Q_k^n]$. Thus, $\Pr(r_{k+1}^{n+1} p_{xy}^{n+1} \geq r_{k+1}^n p_{xy}^n) < \Pr(r_{k+1}^{n+1} p_{xy}^{n+1} < r_{k+1}^n p_{xy}^n)$.

(iv) $Q^{n+1} \leq Q^n \leq 0$: Since $\tau^{n+1} < \tau^n$ and $\frac{Q^{n+1}}{\tau^{n+1}} < \frac{Q^n}{\tau^n}$, we have $p_{xy}^{n+1} < p_{xy}^n$. From (5.23), $Q_k^{n+1} - Q_k^n = \alpha(r_{k+1}^n - Q_k^n) \leq 0$. We obtain $r_{k+1}^{n+1} = r_{k+1}^n \leq Q_k^n \leq 0$. Thus, $r_{k+1}^{n+1} p_{xy}^{n+1} \geq r_{k+1}^n p_{xy}^n$.

(v) $Q^{n+1} \leq 0 \leq Q^n$: Similar to (iv), $p_{xy}^{n+1} < p_{xy}^n$. Using (5.12) and $Q_k^n \geq 0$, we obtain $r_{k+1}^{n+1} = r_{k+1}^n \leq (1 - \frac{1}{\alpha})Q_k^n \leq 0$. Thus, $r_{k+1}^{n+1} p_{xy}^{n+1} \geq r_{k+1}^n p_{xy}^n$.

(vi) $0 \leq Q^{n+1} \leq Q^n$: Similarly, we obtain $p_{xy}^{n+1} < p_{xy}^n$ and $r_{k+1}^n \in [(1 - \frac{1}{\alpha})Q_k^n, Q_k^n]$. Since $Q^{n+1}, Q^n \geq 0$, $\Pr(r_{k+1} < 0) < \Pr(r_{k+1} \geq 0)$. Thus, $\Pr(r_{k+1}^{n+1} p_{xy}^{n+1} \geq r_{k+1}^n p_{xy}^n) < \Pr(r_{k+1}^{n+1} p_{xy}^{n+1} < r_{k+1}^n p_{xy}^n)$.

If $\gamma > 0$, one may replace r_{k+1}^n in (5.23) with $r_{k+1}^n + \gamma f(Q_{k+1}^n)$ and the analysis above still applies to variations of future estimates. Since cases (i)-(vi) are applicable to all (x, y) pairs in any policy π and are equally likely, we conclude that

$$E[R^{n+1}|R^i] = \sum_{\pi \in \Pi} \sum_{k=0}^{L} \sum_{x,y} r_{k+1}^{n+1}(x, y) p_{xy}^{n+1} \tag{5.24}$$

$$\geq \sum_{\pi \in \Pi} \sum_{k=0}^{L} \sum_{x,y} r_{k+1}^{n}(x, y) p_{xy}^{n} \tag{5.25}$$

$$= E[R^n|R^i] = R^n, \quad \forall i \leq n, \tag{5.26}$$

and $\{R^n\}$ is a submartingale. □

We now present the Martingale Convergence Theorem (Theorem 5.14 in [11]) without proof. Its proof can be found in [11], and is omitted here due to limited space.

Lemma 5.4 (Martingale Convergence Theorem)**.** *Let $R^1, R^2, \ldots$ be a submartingale such that $\sup E|R^n| < \infty$, then there exists a random variable (r.v.) R such that $R^n \to R$ almost surely (a.s.) and $E|R| < \infty$.*

Based on Lemma 5.3 and Lemma 5.4, we present the convergence theorem of RLCS.

Theorem 5.2 (Convergence of RLCS Algorithm)**.** *The sequence of expected cumulative rewards R^n, $n = 1, 2, \ldots$ converges to a value R almost surely (a.s.).*

Proof. Since $\{R^n\}$ is a submartingale (Lemma 5.3), by following Lemma 5.4, there exists an r.v. R such that $R^n \to R$ a.s. and $E|R| < \infty$ due to $E|R^n| < \infty$. The convergence of RLCS follows. □

From (5.21), R^n increases with p^n_{xy} according to (5.22). If we set $a = Q^n/\tau^n$ such that $p_i = \frac{e^{a_i}}{\sum_j e^{a_j}}$, we obtain that $\frac{\partial p_i}{\partial a_i} = p_i - p_i^2$ and $\frac{\partial^2 p_i}{\partial a_i^2} = (p_i - p_i^2)(1 - 2p_i)$. p_i is convex if $p_i \in [0, 0.5]$ and concave if $p_i \in [0.5, 1]$. As a result, when τ^n is large at the beginning of learning where exploration takes place, all possible actions i in that state are equally likely and $p_i \in [0, 0.5]$. R^n is convex in this region. On the opposite, when τ^n is close to zero at the end of learning where exploitation takes effect, $p_i = 1$ for the best action i in all states. R^n in this region is concave. The region in between where the transition from exploration to exploitation occurs is, thus, linear. Based on this observation, we show the rate of convergence in the following theorem.

Theorem 5.3 (Rate of Convergence)**.** *The sequence of expected cumulative rewards R^n in RLCS converges sublinearly.*

Proof. Let R^1 be the initial R^n and $R^n = R^*$ in episode $n = N$. The increasing rate of R^n in the linear region between exploration and exploitation can be approximated as $\frac{R^* - R^1}{N-1}$. From $R^n = \frac{R^* - R^1}{N-1}(n-1) + R^1$, we obtain $R^{n+1} - R^* = \frac{n-N+1}{N-1}(R^* - R^1)$

and $R^n - R^* = \frac{n-N}{N-1}(R^* - R^1)$. Using the rate of convergence $\hat{K}$ defined in [73], we have

$$\hat{K} = \limsup_{n\to\infty} \frac{\|R^{n+1} - R^*\|}{\|R^n - R^*\|^\zeta} = \limsup_{n\to\infty} \frac{\|n - N + 1\|}{\|n - N\|} = 1 \tag{5.27}$$

where ζ is the order of convergence and $\zeta = 1$ indicates the convergence of the first order. The same result can be obtained by using (5.16). Since R^* is upper bounded by K^* in (5.16) and $N \gg K^*$ for sufficiently large N, $\Delta R = R^{n+1} - R^n = \frac{R^* - R^1}{N-1} \leq \frac{K^* - R^1}{N-1} \approx 0$. Thus, $\hat{K} = 1$ is obtained in (5.27) with $R^{n+1} \approx R^n$. $\{R^n\}$ is said to converge sublinearly. □

5.4.3 Optimal Stopping Time

In Section 5.3, the RLCS algorithm is introduced to find the optimal solution. However, the number of episodes needed to reach the optimal solution may be large due to the exploration of all state-action pairs in state and action spaces. In this section, we aim to find the optimal stopping time T^* to reduce the total learning time by formulating the problem as Markov optimal stopping with finite horizon N.

Let $\mathcal{F}^n, n = 1, \ldots, N$ be a nondecreasing sequence of sub-σ-algebras of event class $\mathcal{F}$ called a filtration. Consider a set of stopping times $\mathcal{T}_n^N = \{T \in \mathcal{T} | n \leq T \leq N\}, 1 \leq n \leq N$, where $\mathcal{T}$ is the set of all stopping time. Thus, with cumulative reward sequence $\{Y^n\}$, the optimal stopping problem is to find the stopping time T such that the expected cumulative reward is maximized:

$$\sup_{T\in\mathcal{T}} E[Y^T] \tag{5.28}$$

According to the optimal stopping theory [72, 79], the solution to (5.28) can be obtained by backward induction as defined by a sequence of random variables:

$$S_n^N = \max\{Y^n, E[S_{n+1}^N | \mathcal{F}^n]\}, \quad n = N-1, \ldots, 1 \tag{5.29}$$

with $S_N^N = Y^N$. Thus, we stop at $n = T$ if $Y^T \geq E[S_{T+1}^N | \mathcal{F}^T]$ and continue otherwise.

The stopping time is given by

$$T_n^N = \inf\{n \leq T \leq N | S_T^N = Y^T\}, \quad 1 \leq n \leq N. \tag{5.30}$$

However, unlike Y^n, $E[S_{n+1}^N|\mathcal{F}^n]$ is difficult to obtain in each episode n. This is because the probability distributions of the r.v.'s in sequence S_{n+1}^N are unknown in episode n and may be significantly changed owing to the action selections in future episodes $n+1, \ldots, N$. Thus, we use the optimal reward estimate $\tilde{Y}^*$ as the alternative to $E[S_{n+1}^N|\mathcal{F}^n]$. $\tilde{Y}^*$ is the best-known cumulative reward estimate in episode n based on the optimal set of uncorrelated CR users, $\tilde{\mathcal{K}}^*$, obtained from the sequence of average reporting delays, $\mathcal{T}_d^n = \{\bar{t}_{d_i}^n\}$, in an ascending order. By using (5.16), we obtain the following:

$$\tilde{Y}^* = \tilde{K}^* - \frac{\sum_{k=0}^{\tilde{K}^*-1}(\tilde{K}^* - k)\bar{t}_{d_i}^n(k)}{T_{lim}}, i \in \tilde{\mathcal{K}}^* \tag{5.31}$$

where $\tilde{K}^* = |\mathcal{K}^*|$ and $\bar{t}_{d_i}^n(k) \in \mathcal{T}_d^n$ is indexed for the kth uncorrelated CR user in $\tilde{\mathcal{K}}^*$. Thus, (5.29) is reduced to $S_n^N = \max\{Y^n, \tilde{Y}^*\}$ and the optimal stopping time T^* in the RLCS algorithm is the smallest episode number $n = T^*$ whose cumulative reward Y^n is greater than or equal to the optimal reward estimate $\tilde{Y}^*$:

$$T^* = \inf\{n \leq N \mid Y_n \geq \tilde{Y}^*\}. \tag{5.32}$$

In other words, the optimal stopping occurs if the cumulative reward of episode n, Y^n, is greater than or equal to the current best known reward estimate, $\tilde{Y}^*$. Otherwise, the RLCS algorithm continues to find the optimal solution.

5.4.4 Fading Control Channel

As indicated in Section 5.2, FSMC [95] is used to model Rayleigh fading in the control channel for reporting local sensing decisions. In Rayleigh fading, the received SNR γ is exponentially distributed with distribution $f_\Gamma(\gamma) = \frac{1}{\bar{\gamma}}e^{-\gamma/\bar{\gamma}}$, where $\bar{\gamma}$ is the average received SNR. The probability of received SNR γ that stays in SNR

region $[A_j, A_{j+1})$ is the probability of staying in channel state x_j, which is given by $p_j = \int_{A_j}^{A_{j+1}} f_\Gamma(\gamma)\mathrm{d}\gamma = e^{-\frac{A_j}{\bar{\gamma}}} - e^{-\frac{A_{j+1}}{\bar{\gamma}}}$. Since the bit error rate of BPSK modulation in additive white Gaussian noise is $Q(\sqrt{2\gamma})$, the crossover probability of the BSC channel for state x_j is given by [95]

$$\varepsilon_j = \frac{\int_{A_j}^{A_{j+1}} f_\Gamma(\gamma) Q(\sqrt{2\gamma})\mathrm{d}\gamma}{\int_{A_j}^{A_{j+1}} f_\Gamma(\gamma)\mathrm{d}\gamma} = \frac{\gamma_j - \gamma_{j+1}}{p_j} \tag{5.33}$$

where $\gamma_j = e^{-\frac{A_j}{\bar{\gamma}}} Q(\sqrt{2A_j}) + \bar{\gamma}_c\big(1 - Q\big(\frac{\sqrt{2A_j}}{\bar{\gamma}_c}\big)\big)$ and $\bar{\gamma}_c = \sqrt{\frac{\bar{\gamma}}{\bar{\gamma}+1}}$. Hence, the average error probability is

$$P_e = \sum_{j=0}^{J} p_j \varepsilon_j = \gamma_0 - \gamma_\infty = \frac{1}{2}\big(1 - \bar{\gamma}_c\big), \tag{5.34}$$

where the second equality is obtained by canceling intermediate γ_j terms and the last equality is obtained by using $A_0 = 0$ and $A_\infty = \infty$.

Let $\mathcal{D}^n$ be the set of selected CR users for data fusion in episode n and $K^n = |\mathcal{D}^n|$. Let the false alarm probability of CR user i be P_f^i, the detection probability P_d^i, and the average CCC reporting error probability P_e^i in (5.34). For CR user i to report a false alarm in CCC fading, there are two possibilities: 1) a false alarm ($u_i = 1$) is reported and received at FC with no error, and 2) correct local decision $u_i = 0$ is reported and received at FC in error ($u_i = 1$) due to CCC fading. The false alarm probability for local decisions reported by CR user i via fading CCC and perceived by FC is then $P_f^i(1 - P_e^i) + (1 - P_f^i)P_e^i$. Similarly, we can find the probability of CR user i with correct local decisions under H_0 received at FC as $(1-P_f^i)(1-P_e^i)+P_f^i P_e^i$. As a result, the false alarm probability for the cooperative decision in episode n is given by

$$Q_f^n = \sum_{\ell=\lfloor K^n/2 \rfloor}^{K^n} \sum_{\substack{i=1 \\ \mathcal{L}_i \cup \mathcal{M}_i = \mathcal{D}}}^{\binom{K^n}{\ell}} \left(\prod_{\substack{l \in \mathcal{L}_i \\ |\mathcal{L}_i| = \ell}} P_{fe}^l \right) \cdot \left(\prod_{\substack{m \in \mathcal{M}_i \\ |\mathcal{M}_i| = K - \ell}} P_{fe}^m \right) \tag{5.35}$$

where $P_{fe}^l = P_f^l(1 - P_e^l) + (1 - P_f^l)P_e^l$ and $P_{fe}^m = (1 - P_f^m)(1 - P_e^m) + P_f^m P_e^m$ are the probability of CR user l's error reporting and CR user m's correct reporting,

respectively, perceived at the FC, $\mathcal{L}_i$ is the set of the ℓ selected CR users with received false alarms in the "ℓ out of K^n" data fusion rule from the ith combination of all $\binom{K^n}{\ell}$ combinations, and $\mathcal{M}_i = \{\mathcal{D} \backslash \mathcal{L}_i\}$ is the set of the rest of $K^n - \ell$ CR users with correctly received local decisions under H_0 from the ith combination. For the majority decision rule, ℓ ranges from $\lceil K^n/2 \rceil$ to K^n. Similarly, the detection probability for the cooperative decision in episode n, Q_d^n, can be obtained by replacing P_f^l with P_d^l in (5.35) as

$$Q_d^n = \sum_{\ell=\lfloor K^n/2 \rfloor}^{K^n} \sum_{\substack{i=1 \\ \mathcal{L}_i \cup \mathcal{M}_i = \mathcal{D}}}^{\binom{K^n}{\ell}} \left(\prod_{\substack{l \in \mathcal{L}_i \\ |\mathcal{L}_i| = \ell}} P_{de}^l \right) \cdot \left(\prod_{\substack{m \in \mathcal{M}_i \\ |\mathcal{M}_i| = K - \ell}} P_{de}^m \right) \tag{5.36}$$

where $P_{de}^l = P_d^l(1 - P_e^l) + (1 - P_d^l)P_e^l$ and $P_{de}^m = (1 - P_d^m)(1 - P_e^m) + P_d^m P_e^m$.

5.5 Performance Evaluation

In this section, we evaluate the performance of our proposed RLCS scheme by showing the convergence the RLCS algorithm, the improvement of detection probability, and the adaptability to environmental changes.

5.5.1 Simulation Environment

We consider a CR user (the FC) and its 9 neighbors deployed in a 600 m × 600 m square area placed in the first quadrant of the Cartesian coordinate system. A PU with $r_b = 0.3$ and $r_d = 0.2$ at 900 MHz is located at $(0, 0)$. The FC is located at $(500, 500)$ and its neighbors are located within 40 meters of the FC's location. For channel model and sensing parameters, we set $\gamma_{pl} = 3.1$ for path loss, noise uncertainty $\sigma_0 = 6$ dB and lognormal dB-spread $\sigma_1 = 6$ dB in (5.1), decaying coefficient $a = 0.1204$ in (5.2) for urban settings, and local detection threshold $\lambda_{th} = 0.2$ dB for all CR users. The FSMC model for independent Rayleigh fading CCC consists of 1024 SNR regions in which the range of each SNR region is 0.1 dB. We set the CR user transmit power to 20 mW with path loss exponent $\gamma_{pl} = 4.1$. For data fusion at the FC, hard

combinations of local decisions with the majority rule are used. With these settings, the CR users are approximately located at the boundary of the protected region of the PU. At this border location, the received power is close to the noise floor set to -101 dBm. Thus, cooperative sensing is essential for CR users to improve their detection performance.

Table 5.1 lists one of random deployments of an agent (ID 0) and 9 cooperating CR users. The cooperating CR users are randomly deployed around the agent. The PU not shown in the Table is located at the origin $(0, 0)$. From their coordinates, one can find that CR users 1, 7, and 8 are strongly correlated. CR users 6 and 9 are correlated as well as CR users 3 and 5. In addition, each cooperating CR user has different sensing report delay and sensing schedule, and may change its location over time. For illustration purpose, the location, sensing report delay, and schedule priority of these CR users are assumed to be fixed in this scenario. From the table, the order of CR user sensing report delay is $\{3, 6, 8, 7, 2, 1, 5, 4, 9\}$ while the schedule priority is $\{6, 4, 5, 3, 9, 8, 2, 1, 7\}$. For example, CR user 3 has the smallest sensing delay among all cooperating CR users and CR user 6, if selected, is the first CR user to report local decisions. By using RLCS, we can obtain the optimal solution including the optimal set of cooperating CR users in the sensing report order: $\{6, 4, 3, 8, 2\}$ with the total sensing report delay 29.58ms. Compared to the full cooperation by all 9 neighbors, which requires total delay 79.35ms, a 63% saving of cooperative sensing time is obtained in this case. It is also more energy-efficient (up to 44% saving) since only selected neighbors (optimal 5 $<$ total 9 CR users) are required to perform local sensing and report their results for each round of cooperation. Thus, the detection performance, sensing delay, and energy efficiency of RL-based cooperative sensing are all considerably improved from the cooperation by all neighbors under correlated shadowing.

Table 5.1: Location, Reporting Delay, and Schedule Priority of CR Users

ID	x-coord	y-coord	Delay (ms)	Priority
0	500	500	-	-
1	485	514	11.38	8
2	502	494	4.67	7
3	504	512	2.72	4
4	489	486	14.81	2
5	506	514	12.80	3
6	501	532	3.15	1
7	486	515	4.60	9
8	487	513	4.24	6
9	503	534	20.99	5

5.5.2 Convergence of RLCS Algorithm

Figure 5.4 shows the expected cumulative rewards with $N = 1000$, $\tau_0 = 1$, and $\tau_N = 0.01$, for Q-learning, Sarsa, and Action-Critic TD learning strategies over 1000 runs. Interested readers may refer to Appendix B for an introduction to these TD learning algorithms. The sublinear convergence of RLCS (both with and without optimal stopping) is evident. The maximum value R^* is obtained by (5.16). For both cases, Q-learning converges to the optimal value while the other two settle for a suboptimal value due to more exploitation than exploration in early stages ($n < 500$). All three methods show significant improvement in R_n with optimal stopping.

5.5.3 Detection Performance

Figure 5.5 shows the improvement of detection performance (both Q_d and Q_f) under Poisson and bursty PU traffic during the RLCS process. Q_d and Q_f are averaged over the most recent 500 cooperative decisions at the FC. The detection performance of full cooperation case (FCS) is approximately the same for all episodes. It is evident that Q_d of RLCS is gradually improved and reaches above 0.9 after 3000 episodes, significant improvement over FCS. The initial large Q_f of RLCS is attributed to non-optimal CR user selections at the beginning of learning during the exploration phase. However, Q_f is constantly decreasing and considerably reduced to 0.025 to be

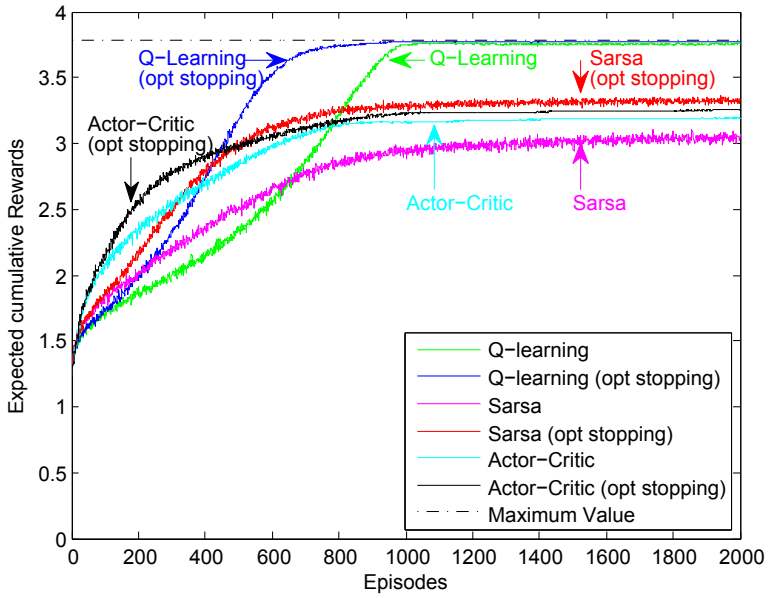


Figure 5.4: Expected Cumulative Rewards of RL-Based Cooperative Sensing.

comparable to Q_f of FCS at the end. Thus, with RLCS, the detection performance improves as soon as the learning from the environment takes effect. Figure 5.6 shows the receiver operating characteristic (ROC) curves of FCS and RLCS in correlated shadowing and possible user movement and CCC fading. We see that the cooperative gain achieved by FCS in independent shadowing is compromised by correlated shadowing with Q_d dropping from 0.97 to 0.85 at $Q_f = 0.1$. The detection performance of full cooperation in independent shadowing is attainable only when all cooperating CR users are uncorrelated. However, with RLCS, Q_d is increased to 0.91. Hence, RLCS scheme is effective in combating correlated shadowing.

5.5.4 Adaptability to Environmental Change

With the learning capability, the RLCS algorithm is able to adapt to changes in the environment. In this subsection, we evaluate the adaptability of RLCS based on the changes of PU activity, user location, user reliability, and fading control channel.

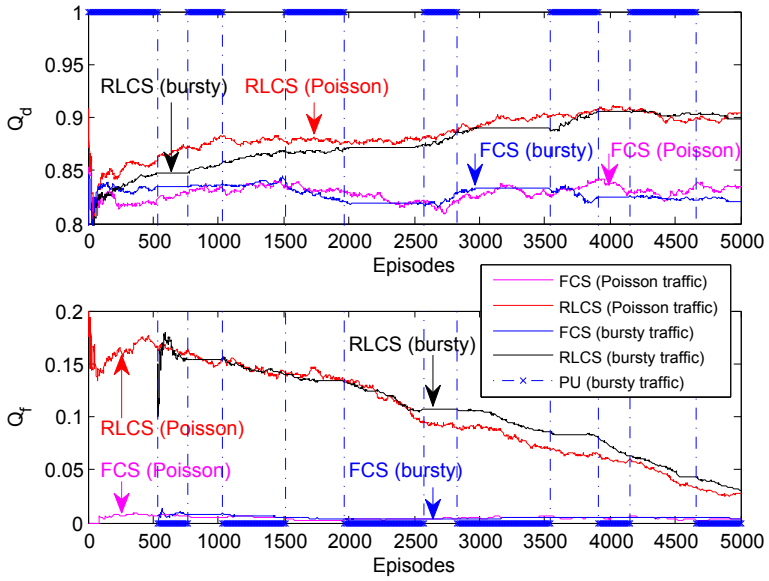


Figure 5.5: Improvement of Q_d/Q_f during RLCS and Adaptability to Random and Bursty PU Traffic.

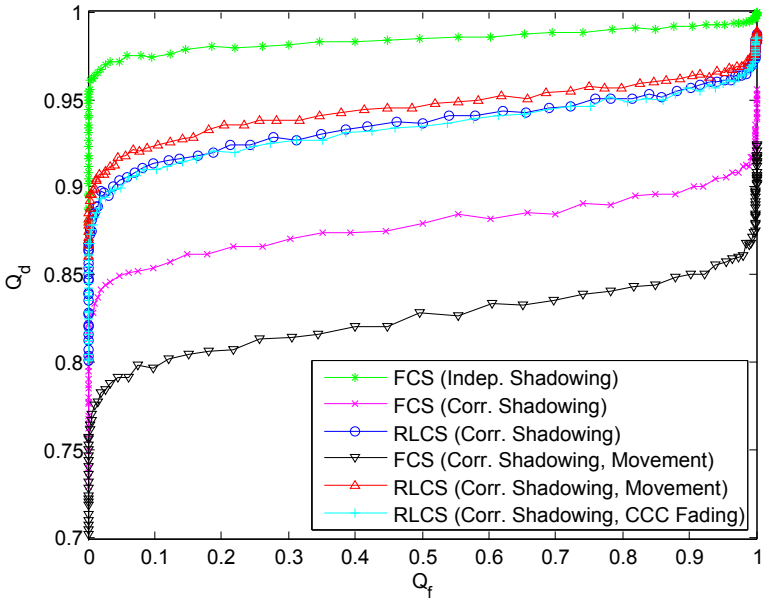


Figure 5.6: ROC of FCS and RLCS in Correlated Shadowing with Possible User Movement and Control Channel Fading.

5.5.4.1 PU Activity Changes

Figure 5.5 shows adaptability to PU activity changes for different PU traffic types in addition to constant improvement in detection performance. To manifest the effect

of PU activity changes on detection performance, we generate bursty PU traffic by staying in either PU state with high probability for a period that spans a random number of episodes, and toggling the ON/OFF states with low probability. As seen in the figure, Q_d is improved mostly during the period of active PU while the Q_f is primarily improved during the period of no PU activity. For this reason, $Q_f = 0$ during the first 500 episodes. Thus, RLCS is adaptable to PU activities and consistently improves detection performance in arbitrary PU traffic patterns.

5.5.4.2 User Movement

In this test scenario, a CR user with independent observations at original location moves to a new location (distance 33.12 m) at the pedestrian speed 1.25 m/s. Since it takes 26.5 s to reach the destination, the movement spans 133 RLCS episodes for cooperative sensing period of 200 ms. Since this movement incurs correlation with other CR users, the optimal solution is changed accordingly when the algorithm converges. Figure 5.6 shows the ROC curve before and after the movement for both FCS and RLCS. For $Q_f = 0.1$, Q_d of FCS drops from 0.85 to below 0.8 after the movement while Q_d of RLCS is slightly improved due to the selection of all uncorrelated CR users. This shows the capability of RLCS adapting to user movement while maintaining detection performance.

5.5.4.3 User Reliability

Figure 5.7 shows the KL distance curve of an unreliable CR user, the average KL distance values of all reliable users, and the detection threshold $\delta_{KL} = \mu_\delta + 2\sigma_\delta$ over 5000 episodes. The low KL values before episode 1000 indicate that the CR user is normally a reliable user. Its KL value is dramatically increased after the user becomes unreliable in episode 1000. It is detected and removed from the set of cooperation in episode 1391 when its KL value exceeds the threshold δ_{KL}. When the CR user becomes reliable again, its KL distance is gradually reduced. After its KL distance

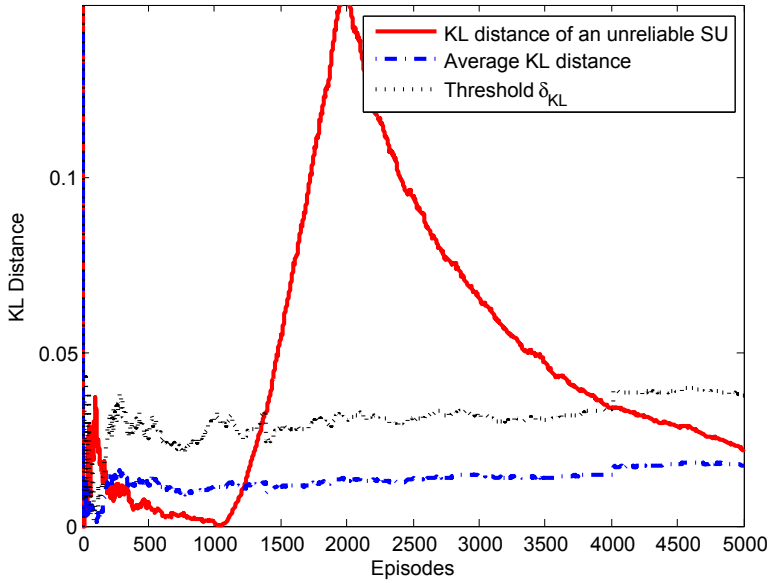


Figure 5.7: Average and User KL Distance Values for Detection of Unreliable Users.

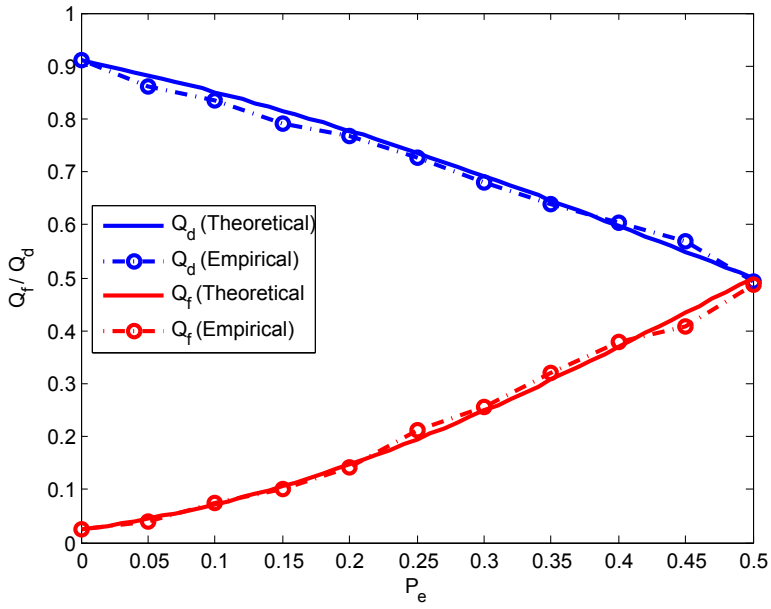


Figure 5.8: Theoretical and Empirical Detection Performance (Q_d/Q_f) versus Average Error Probability (P_e) on Fading Control Channel.

meets the threshold in episode 4016, the user may be selected by the FC again for cooperation.

5.5.4.4 Fading Control Channel

We first compare the detection performance obtained by simulations with that obtained by using (5.35) and (5.36). Local P_d and P_f are set to 0.7633 and 0.1466, respectively, for all CR users. This corresponds to $Q_d = 0.91$ and $Q_f = 0.025$, respectively, with $P_e = 0$. The results are obtained in episode 5000 when the optimal solution is reached, and are averaged over 10 runs. Figure 5.8 shows the theoretical and empirical detection performance versus P_e ranging from 0 to 0.5 on fading control channel. The simulation results follow the theoretical curves closely for both Q_d and Q_f. Figure 5.6 also shows the ROC curve of RLCS with fading CCC versus perfect CCC. The detection performance of the fading CCC is similar to that of the perfect CCC case with slight degradation ($Q_f = 0.0255, Q_d = 0.9027$). Thus, RLCS can effectively maintain the detection performance with reporting in fading control channel.

CHAPTER VI

JAMMING-RESILIENT CONTROL CHANNELS FOR INTRUSION DEFENSE

6.1 Motivation

In the previous two chapters, we introduce ERCC and RLCS to address issues of the responsiveness to PU activities and the robustness to channel impairments, respectively. In this chapter, we aim to address the issue of resilience to jamming attacks. As in any wireless network, security attack can cause severe damage to CR ad hoc networks. Among all types of security attacks, control channel jamming is known to be an efficient and effective way for intelligent attackers to result in DoS. Regardless of the challenges of CCC establishment and recovery in licensed bands, control channel jamming poses additional challenges to control channel reliability and security in CR ad hoc networks.

Common approaches to combating control channel jamming in wireless networks are duplicate control messages on multiple control channels [15,86], random key distribution to hide CCC locations [15,86,87], and channel hopping [47]. To reduce control overhead, the duplicate control channel allocations should be minimized. Moreover, as pointed out in [94], channel hopping should adapt to attackers' jamming strategies because a fixed channel hopping sequence may be easily deciphered by intelligent attackers. Thus, in response to attacker's intelligence, CR users must be equipped with learning capability to develop their own defense strategies [51].

The interactions between CR users and attackers are commonly modeled as a stochastic zero-sum game [50, 94, 106] since CR users and attackers generally have opposite goals. In these approaches, PU activities govern states of the game and

state transitions, and sensing errors are generally ignored for simplicity. In [50], the Nash equilibrium strategy is obtained for the one-stage game, while the optimal attacking strategy is obtained for the multi-stage case. The latter is achieved by fixing CR user's strategy and converting the problem to the framework of the single-player partially observable Markov decision process (POMDP). [106] shows that CR users can combat jamming by increasing the number of unoccupied channels that can be observed. However, this capability is limited by PU activity and channel availability. In [94], minimax-Q learning is used by CR users to find the optimal anti-jamming channel selection policy. Although CR users' actions consist of separate selections of control and data channels, attackers in this work, like those in [50] and [106], do not exclusively target at jamming control channels and consider the impact of spectrum sensing errors on CR users and attackers.

To address these challenges, we model the interactions of intelligent attackers and CR users in their jamming regions under the impact of PU activities as a stochastic general-sum game, called jamming-resilient control channel (JRCC) game, and select the optimal control channel allocation strategies by using an enhanced multiagent reinforcement learning (MARL) algorithm, called JRCC algorithm. The objective of the game is to find the optimal control channel allocation strategy for CR users to combat jamming attacks by using JRCC algorithm. The optimal control channel allocation policy is obtained by enabling the communications among CR users to facilitate CCC allocations and the adaptation to PU activity to achieve the Nash equilibrium in the game.

Figure 6.1 illustrates the JRCC game with a PU and a CR ad hoc network attacked by spatially distributed attackers. The circle centered at each attacker is its effective jamming region. It is for attackers' benefits such as energy saving to minimize the overlapping areas of the jamming regions. As a result, we focus on one jamming region where one attacker jams the control channels established between CR

users inside its jamming region in this work. We demonstrate that the effectiveness of jamming-resilient CCC allocations can be improved by the proposed cooperative intrusion defense strategies that utilize the cooperation of CR users such as action strategy coordination with variable learning rates, best-effort cooperative sensing, and scalable CR user deployment. Our enhanced JRCC algorithm combats jamming under a wide range of PU activities and spectrum sensing errors by exploiting the advantages of both Policy Hill-Climbing (PHC) and PHC-Win-or-Learn-Fast (PHC-WoLF) multiagent reinforcement learning (MARL) algorithms [10], and outperforms these original MARL algorithms.

In Section 6.2, we discuss the JRCC system model and assumptions. In Section 6.3, we describe the JRCC game, its gradient dynamics, the convergence of the game, and the proposed JRCC algorithm. In Section 6.4, we analyze the impact of PU activity and spectrum sensing errors on jamming resilience of CR users and jamming strength of the attacker. In Section 6.5, we present and analyze the proposed cooperative intrusion defense strategies for improving jamming resilience. In Section 6.6, we evaluate the performance of jamming resilience of CR users and jamming strength of the attacker in the JRCC game. In Appendix A, we tabulate the variables and notations used in this Chapter in Table A.5 and A.6 for reference. Our contributions of this study are summarized as follows:

- We model the interactions among CR users and the attacker under the impact of PU activities and spectrum sensing errors as a stochastic general-sum game, analyze its gradient dynamics, and show Nash equilibriums and the convergence of the game.

- We propose the JRCC algorithm with cooperative intrusion defense strategies as optimal strategies to significantly enhance jamming resilience of CR users by exploiting the advantages of MARL algorithm and CR user cooperation.

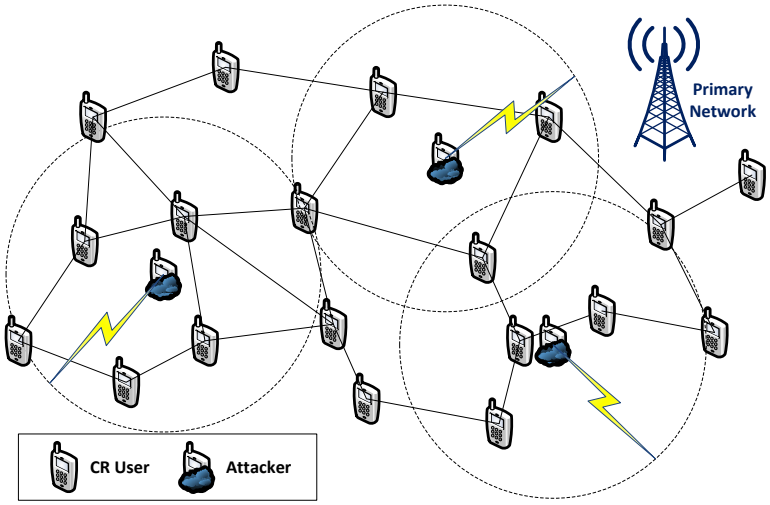


Figure 6.1: Jamming-Resilient Control Channel Game.

- We analyze the impact of PU activities and the effect of spectrum sensing errors on CR users and attackers, and the effectiveness of intrusion defense strategies including action coordination, best-effort cooperative sensing, and CR user deployment density and scalability.

6.2 JRCC System Model

The JRCC system model consists of a primary network model, a CR ad hoc network model, and a jamming attack model. The interactions among these users are modeled as a general-sum stochastic game. Interested readers may refer to Appendix C for a summary of stochastic game. The JRCC system model is described in detail as follows.

Primary Network Model: The primary network $\mathcal{P}$ consists of N_p PUs and a set of N_p licensed channels, $\mathcal{N}_p$. These channels are available for opportunistic access by CR users in an area larger than the jamming region $\mathfrak{A}$. Each licensed channel $i \in \mathcal{N}_p$ is occupied by one PU, P_i, whose activity follows the two-state birth-death process with birth rate r_{b_i} and death rate r_{d_i}. The departures and the arrivals of a PU on channel i follow a Poisson process with exponentially distributed inter-arrival

time. Thus, each channel i has two states, PU active (ON) state and PU inactive (OFF) state, with transition probabilities: r_{b_i} (OFF to ON) and r_{d_i} (ON to OFF). The long-term average probabilities of PU active (P_{on}) and inactive (P_{off}) on channel i are $\frac{r_{b_i}}{r_{b_i}+r_{d_i}}$ and $\frac{r_{d_i}}{r_{b_i}+r_{d_i}}$, respectively. PU transmissions are assumed to be time-slotted. Thus, the players of the game including CR users and attackers need to periodically sense licensed channels according to the schedule of the primary network. Since the sensing operations are subject to errors, we assume that CR users and attackers are required to satisfy the detection requirements in terms of probability of false alarm P_f and probability of miss detection P_m to limit the interference with PUs under a tolerable level.

CR Ad Hoc Network Model: A group of K CR users, $\mathcal{K}$, are uniformly deployed with density D_K in the jamming region $\mathfrak{A}$ and opportunistically access N_p licensed channels with the objective to establish valid control channels for rendezvous. We assume that CR users are able to observe all the channel states by wideband spectrum sensing. However, CR users may choose a subset of channels for transmission due to energy-saving consideration. As a result, after observing the channel states by spectrum sensing, CR user $K_i \in \mathcal{K}$ selects N_k available channels as control channels for control transmission. Limited by hardware capability, CR users can only sense or transmit on $N_s \leq N_p$ licensed channels each time, and select a subset of available channels, $\mathcal{N}_k \subseteq \mathcal{N}_p$, as control channels, and transmit the same control messages on those selected channels, where $N_k = |\mathcal{N}_k| \leq N_s$. We assume that all control messages are encrypted and are unable to be deciphered by eavesdropping attackers during the period of the game. After the rendezvous and control message exchange on these CCCs, the CR user pair can use the in-band CCCs for channel negotiation or data transmission.

Jamming Attack Model: A group of J intelligent attackers, $\mathcal{J}$, equipped with similar hardware capability as CR users are uniformly deployed with density D_K in

$\mathfrak{A}$. The main objective of attackers is to disrupt CR control transmission. Similar to CR users, attackers can only sense or transmit on $N_s \leq N_p$ licensed channels each time. Each attacker $j \in \mathcal{J}$ selects a subset of channels, $\mathcal{N}_j \subseteq \mathcal{N}_p$ and $N_j = |\mathcal{N}_j| \leq N_s$, and transmits jamming signals on those selected channels. We assume that attackers will make efforts to avoid interfering with PUs to save their energy and avoid being exposed to PUs unless the interference is caused by the limitation of sensing hardware. To achieve this, attackers are also required to satisfy the requirements of P_f and P_m such that they appear to PUs as CR users. Moreover, we assume that attackers do not behave like PUs to occupy the channels because CR users can easily detect these attackers with spectrum sensing and select other channels for control transmission. We further assume that attackers are unable to effectively launch control channel jamming attacks after CCCs are established because such attacks require the prior knowledge about CR users and the in-band CCCs are also used for data transmission. For these reasons, security attacks such as PU emulation attacks [17] and node capture attacks [87] are beyond the scope of this work.

6.3 Multiagent Jamming-Resilient Control Channel Game

In this section, we discuss the JRCC game that models the interactions among players. We first introduce the definitions of CCCs, control links, jamming resilience, and jamming strength. We then present the elements of the JRCC game, gradient dynamics analysis, and the proposed JRCC algorithm.

6.3.1 Jamming Resilience and Jamming Strength

We first define valid CCCs and jammed CCCs in JRCC game. Let P_i and P_j be the transmission power of CR user i and attacker j, respectively. If CR user $i \in \mathcal{K}$ sends control messages to CR user $k \in \mathcal{K}$, $i \neq k$, in a PU-free channel subject to jamming attacks from attacker j. The signal to interference plus noise ratio (SINR) γ_k at CR

user k is given by

$$\gamma_k = \frac{|h_{ik}|^2 P_i}{\sigma^2 + \sum_j |h_{jk}|^2 P_j} \tag{6.1}$$

where h_{ik} and h_{jk} are the channel gain between CR users i and k, and between attacker j and CR user k, respectively, and σ^2 is the noise power. Thus, we have the following definitions of valid and jammed CCCs:

Definition 6.1. *(Valid Common Control Channel) A valid CCC c_v is a control channel established between two CR users $i, k \in \mathcal{K}$ that satisfies (i) $c_v \notin \mathcal{D}_p$, (ii) $c_v \in (\mathcal{N}_i \cap \mathcal{N}_k)$, and (iii) $\gamma_i, \gamma_k \geq \gamma_{th}$ where $\mathcal{D}_p$ is the set of channels occupied by PUs and γ_{th} is the SINR threshold for decoding control messages.*

Based on this definition, we can define a *selected CCC* c_s between two CR users as a control channel that satisfies condition (ii), but may not satisfy conditions (i) and (iii). In the perfect spectrum sensing cases with no spectrum sensing errors, a selected CCC also satisfies condition (i) in Definition 6.1.

Definition 6.2. *(Jammed Common Control Channel) A jammed CCC c_j is a control channel selected by CR users $i, k \in \mathcal{K}$ and jammed by attacker $j \in \mathcal{J}$ that satisfies (i) $c_j \notin \mathcal{D}_p$, (ii) $c_j \in \{\mathcal{N}_i \cap \mathcal{N}_j \cap \mathcal{N}_k\}$, and (iii) $\gamma_i < \gamma_{th}$ or $\gamma_k < \gamma_{th}$.*

Let U_c, N_c, P_c, and J_c be the number of valid CCCs, selected CCCs, PU-occupied selected CCC, and jammed CCCs, respectively, between two CR users. We can find U_c in terms of N_c, N_c, and J_c as

$$U_c = \begin{cases} N_c - P_c - J_c, & \text{if } N_c > P_c + J_c, \\ 0, & \text{if } N_c = 0 \text{ or } N_c = P_c + J_c. \end{cases} \tag{6.2}$$

We can now define valid, invalid, and jammed control links as follows.

Definition 6.3. *(Control Link) A control link between two CR users $i, k \in \mathcal{K}$ is (i) valid if $U_c > 0$, (ii) invalid if $U_c = 0$ and $J_c = 0$, and (iii) jammed if $U_c = 0$ and $J_c > 0$.*

To evaluate the performance of CR users and attackers, we are interested in knowing how many valid and jammed CCC links in the jamming region in each stage and in the long run. Thus, we define the jamming resilience of a CR ad hoc network and the jamming strength of an attacker as follows.

Definition 6.4 (Jamming Resilience)**.** *The jamming resilience of CR users in CR ad hoc network $\mathcal{K}$ is defined as the long-term average ratio of the number of established control links to the number of all possible control links in jamming region $\mathfrak{A}$ given by*

$$\psi(\mathfrak{A}) = \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \frac{C_{\mathcal{K}}^t}{L_{\mathcal{K}}^t}, \tag{6.3}$$

where $C_{\mathcal{K}}^t$ is the number of valid CCC links established in stage t and $L_{\mathcal{K}}^t$ is the number of possible CCC links for establishment in the jamming region of the attacker j in stage t.

$L_{\mathcal{K}}^t$ is given by $L_{\mathcal{K}}^t = \binom{K}{2}$ if K CR users are located in the jamming region and within the transmission rage of each other.

Definition 6.5 (Jamming Strength)**.** *The jamming strength of attackers is defined as the long-term average ratio of the number of jammed control links to the number of all possible control links in jamming region $\mathfrak{A}$ given by*

$$\zeta(\mathfrak{A}) = \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \frac{D_{\mathcal{K}}^t}{L_{\mathcal{K}}^t}, \tag{6.4}$$

where $D_{\mathcal{K}}^t$ is the number of jammed CCC links in stage t.

Note that $C_{\mathcal{K}}^t$ is generally not equal to $L_{\mathcal{K}}^t - D_{\mathcal{K}}^t$ because an invalid CCC link can be caused by something other than jamming such as spectrum sensing errors or two neighboring CR users choosing different sets of channels. Thus, $\psi(\mathfrak{A}) \neq 1 - \zeta(\mathfrak{A})$.

6.3.2 Elements of JRCC Game

The JRCC game Γ is a general-sum stochastic game with N players including K CR users and J attackers (in our case, $N = K + J$ where $J = 1$), which can be represented

by the 4-tuples $\Gamma = \langle \mathcal{S}, \mathcal{A}, f_p, f_R \rangle$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, f_p is the transition probability function, and f_R is the reward function vector as follows.

States: The states of the JRCC game are the channel states of N_p licensed channels determined by the PU activity of primary network $\mathcal{P}$. The state space is then $\mathcal{S} = \times_i \mathcal{S}_i$ where $\mathcal{S}_i = \{0, 1\}$ is the set of channel states of channel i, $i = 1, \ldots, N_p$. If the state of channel i in stage t is denoted by s_i^t, the state of the game at stage t is $\mathbf{s}^t = \{s_1^t, \ldots, s_{N_p}^t\}$. Here $s_i^t = 1$ if channel i is occupied by PU P_i at stage t and $s_i^t = 0$ if channel i is available. Thus, there are total 2^{N_p} states in the game.

Actions: The action space $\mathcal{A}$ is the set of joint action spaces of all states given by $\mathcal{A} = \{\mathcal{A}(1), \ldots, \mathcal{A}(2^{N_p})\}$ where $\mathcal{A}(s)$ is the joint action space of all players in state s. $\mathcal{A}(s)$ is then given by $\mathcal{A}(s) = \times_n \mathcal{A}_n(s)$ where $\mathcal{A}_n(s)$ is the set of actions available to player n in state s. An action is a subset of all available channels selected by a player for transmission. Each player n selects an action a_n^t in each stage t according to a mixed strategy $\pi_n^t(s) \in \Delta(\mathcal{A}_n(s))$, where $\Delta(x)$ is the probability distribution of x, after observing state s by spectrum sensing. The number of actions available in state s is then $M = |\mathcal{A}_n(s)| = \sum_{i=0}^{L(s)} \binom{C(s)}{i}$, where $C(s)$ is the number of available channels in state s and $L(s)$ is the minimum number of $C(s)$ and N_s.

Transition Probabilities: The transition probability function is defined as $f_p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$. Since the state transitions are governed by PU activity and all channels are independent, the state transition probability is given by $\mathrm{P}(\mathbf{s}^{t+1} \mid \mathbf{s}^t) = \prod_{i=1}^{N_p} \mathrm{P}(s_i^{t+1} = j \mid s_i^t = k)$, $j, k \in \{0, 1\}$ where $\mathrm{P}(s_i^{t+1} \mid s_i^t)$ is the probability of state transitions from state s_i^t to s_i^{t+1} on channel i, which are $\mathrm{P}(s_i^{t+1} = 1 \mid s_i^t = 0) = r_{b_i}$, $\mathrm{P}(s_i^{t+1} = 0 \mid s_i^t = 0) = 1 - r_{b_i}$, $\mathrm{P}(s_i^{t+1} = 0 \mid s_i^t = 1) = r_{d_i}$, and $\mathrm{P}(s_i^{t+1} = 1 \mid s_i^t = 1) = 1 - r_{d_i}$, depending on the PU ON/OFF status of the given state.

Rewards: The reward function vector is defined as $f_R = \{f_{R_n}\}_1^N$ where $f_{R_n} = \mathcal{S} \times \mathcal{A} \to \mathbf{R}$ is the reward function of player n. In each stage t, player n receives reward r_n^t for the selected action a_n^t in state s. The reward is the result of the joint

action $(a_1^t, \ldots, a_N^t)$ selected by the joint strategy $(\pi_1^t, \ldots, \pi_N^t)$. Thus, the objective of the game for each player n is to find the optimal strategy π_n^* to maximize its average reward R^{π_n} given by

$$R^{\pi_n} = \liminf_{T\to\infty} \left[\frac{1}{T} \sum_{t=1}^{T} R_n^t(s, \pi_1, \ldots, \pi_N) \right] \tag{6.5}$$

$$= \liminf_{T\to\infty} \left[\frac{1}{T} \sum_{t=1}^{T} E^{s,\pi_1^t,\ldots,\pi_N^t} \left[r_n^t(s, \pi_1^t, \ldots, \pi_N^t) \right] \right] \tag{6.6}$$

where R_n^t is the expected reward in stage t and r_n^t is player n's immediate reward received in stage t. Since CR users are rewarded for establishing valid CCC links while attackers are rewarded for successfully jamming selected CCCs, the immediate reward of CR user k in stage t is defined as

$$r_k^t = \frac{1}{B_k} \sum_{i=1}^{B_k} \mathcal{I}\{C_{k,i} > 0\}, \tag{6.7}$$

where B_k is the number of CR user k's neighbors, $\mathcal{I}\{x\}$ is the indicator function that equals 1 if x is true and 0 otherwise, and $C_{k,j}$ is the number of valid CCCs allocated to the i-th neighbor, and the immediate reward of attacker j in stage t is defined as

$$r_j^t = \frac{1}{L_\mathcal{K}} \sum_{i=1}^{L_\mathcal{K}} \mathcal{I}\{D_{j,i} > 0 \text{ and } C_{k,i} = 0\}, \tag{6.8}$$

where $L_\mathcal{K}$ is the number of CR links subject to jamming in the jamming region of attacker j, and $D_{j,i}$ is the number of jammed CCCs in the i-th control link.

6.3.3 Gradient Dynamics Analysis

In the JRCC game, the interactions among all players can be modeled as a nonlinear dynamical system in which the dynamics lies in the gradient of the joint strategy. Similar to [10, 80], we examine the dynamics of the JRCC game using the gradient ascent and show that the players' strategies or expected payoffs will converge. Unlike [10, 80], we focus on the dynamics of an N-player JRCC game with K CR users and one attacker. For simplicity, we assume perfect spectrum senisng and full observations of PU states with no state change.

In stage t of the game, each player n, $n = 1, \ldots, N$, chooses action $a^t_{n,m} \in \mathcal{A}_n(s)$, $m = 1, \ldots, M$ in state s where $M = |\mathcal{A}_n(s)|$. This indicates that player n selects the m-th subset of PU-free channels for CCC allocation if the player is a CR user or for jamming if an attacker. Let $\pi^t_n(s) = \{\pi^t_{n,m} \in [0,1] : \sum_{m=1}^{M} \pi^t_{n,m} = 1\}$ be player n's action selection strategy in stage t where $\pi^t_{n,m}$ is the probability of choosing action $a^t_{n,m}$. According to the joint strategy $(\pi^t_1, \ldots, \pi^t_N)$, the immediate reward of player n choosing action $a^t_{n,m}$ in stage t is $r^t_{n,m}(\pi^t_1, \ldots, \pi^t_N)$. Thus, the expected reward R^t_n of player n can be expressed as

$$R_n(\pi^t_1, \ldots, \pi^t_N) = \sum_{m=1}^{M} \pi^t_{n,m} r^t_{n,m}(\pi^t_1, \ldots, \pi^t_N). \tag{6.9}$$

The gradient ascent algorithm provides the mechanism for a player to achieve the optimal solution by iteratively adjusting its strategy with a sufficiently small step size. In the gradient ascent using variable learning rates [10], the changes in expected rewards can be expressed as iterative strategy update rules as follows:

$$\mathbf{x}^{t+1}_n = \mathbf{x}^t_n + \alpha^t \delta^t_n \frac{\partial R_n(\pi^t_1, \ldots, \pi^t_N)}{\partial \mathbf{x}^t_n}, \quad n = 1, \ldots, N \tag{6.10}$$

where $\mathbf{x}^t_n = [\pi^t_{n,1}, \ldots, \pi^t_{n,M-1}]^{\mathrm{T}}$ with $\pi^t_{n,M} = 1 - \sum_{m=1}^{M-1} \pi^t_{n,m}$, $\delta^t_n > 0$ are the learning rates, and $(\alpha^t \delta^t_n)$ are the step sizes for updating strategy $\mathbf{x}^t_n$ in stage t. $\frac{\partial R_n}{\partial \mathbf{x}^t_n}$ represent the changes in player n's expected reward in response to the changes in the strategy $\mathbf{x}^t_n$ in the direction of the gradient. They are obtained by taking the partial derivatives of each player's expected reward with respect to its strategy. When the step sizes are sufficiently small, the dynamics of the strategy changes can be formulated as a constrained nonlinear affine dynamical system with differential equations defined as

$$\dot{\mathbf{x}} = \boldsymbol{\Delta}(\mathbf{A}\mathbf{x} + \mathbf{b} + \mathbf{c}(\mathbf{x})) \tag{6.11}$$

subject to the unit-hypercube constraints:

$$\mathbf{x} \in [0,1]^{N(M-1)}, \tag{6.12}$$

where $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^{\mathrm{T}}$, $\mathbf{\Delta} = [\delta_1, \ldots, \delta_N]\mathbf{I}_{N(M-1)}$, $\mathbf{A}_{N(M-1)\times N(M-1)}$ and $\mathbf{b}_{N(M-1)\times 1}$ are matrices whose elements are the functions of rewards $r^t_{n,m}$, and $\mathbf{c}(\mathbf{x})_{N(M-1)\times 1}$ are the remainder functions of $\mathbf{x}$ that contains higher-order products of $\mathbf{x}_1, \ldots, \mathbf{x}_N$. The constraints limit the strategies inside the unit hypercube because every strategy in the N-tuples is a probability distribution represented by a point in the $(M-1)$-simplex.

The system can be linearized at a fixed point $\mathbf{x}^*$ if it has a solution $\mathbf{x}^*$ [4]. If we let $r = \|\mathbf{x} - \mathbf{x}^*\|_2$, $\mathbf{c}(\mathbf{x})/r$ approach $\mathbf{0}$ faster than r as $r \to 0$. Combined with the change of variable $\mathbf{y} = \mathbf{x} - \mathbf{x}^*$, we obtain the homogeneous linear system:

$$\dot{\mathbf{y}} = \mathbf{\Delta}\mathbf{J}\mathbf{y} \tag{6.13}$$

subject to the unit-hypercube constraints:

$$\mathbf{y} \in [0, 1]^{N(M-1)}, \tag{6.14}$$

where $\mathbf{J} = \mathbf{J}_{\mathbf{F}}|_{\mathbf{x}=(\mathbf{x}^*_1,\ldots,\mathbf{x}^*_N)}$, and $\mathbf{J}_{\mathbf{F}}$ is the partial derivative (Jacobian matrix) of $\mathbf{F}(\mathbf{x}) = [\mathbf{F}_1, \ldots, \mathbf{F}_{N(M-1)}]^{\mathrm{T}} = \mathbf{A}\mathbf{x} + \mathbf{b} + \mathbf{c}(\mathbf{x})$ given by

$$\mathbf{J}_{\mathbf{F}} = \begin{pmatrix} \frac{\partial \mathbf{F}_1}{\partial \pi_{1,1}} & \cdots & \frac{\partial \mathbf{F}_1}{\partial \pi_{N,M-1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{F}_{N(M-1)}}{\partial \pi_{1,1}} & \cdots & \frac{\partial \mathbf{F}_{N(M-1)}}{\partial \pi_{N,M-1}} \end{pmatrix}$$

The phase portraits of the non-linear system and its linearized system are considered qualitatively equivalent in the neighborhood of $\mathbf{x}^*$. Based on the analysis of gradient dynamics, we obtain the following theorem.

Theorem 6.1 (Convergence Theorem of JRCC Game)**.** *For the N-player general-sum JRCC game, if the players follow the gradient ascent algorithm with variable learning rates and a sufficiently small step size, the strategy N-tuples $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ will converge to a Nash equilibrium or the expected rewards of the players will converge to the expected rewards of a Nash equilibrium in the limit.*

Proof. We examine the coefficient matrix $\mathbf{J}$ of the linear dynamical system (6.13) with the constraints (6.14), and show that the strategies will either converge to the fixed points of the system inside the unit hypercube or the expected rewards of the strategies will converge to those of a Nash point on the boundary of the hypercube. Since the variable learning rates in $\mathbf{\Delta}$ have no effect on the direction of the gradient, we focus on the eigenanalysis of $\mathbf{J}$ in the following two cases.

1) $\mathbf{J}$ *is singular:* In this case, the system is neutrally stable and the trajectories in the phase portrait exhibit periodic patterns and the strategy N-tuples are periodic functions of time. Thus, there exists a periodic point $\mathbf{x}^*$ such that

$$\mathbf{P}^t(\mathbf{x}^*) = \mathbf{x}^*, \quad \forall t \in \mathbb{Z}^+, \tag{6.15}$$

where $\mathbf{P}$ is the first recurrence map known as Poincaré map. Since this periodicity in the strategies is predictable and not desired by either CR users or the attacker in the JRCC game, CR users and the attacker will enforce the system to stay away from neutrally stable states in order to make their strategies unpredictable. Players in these situations may reset their strategies to the initial or uniform distribution in order to avoid the periodicity and maximize the uncertainty of the strategies.

2) $\mathbf{J}$ *is nonsingular:* In this case, $\mathbf{J}$ is invertible and all the eigenvalues of $\mathbf{J}$ have nonzero real part. The system has hyperbolic fixed points: the phase portraits of the non-linear system and its linearization are qualitatively equivalent in the neighborhood of the fixed points. Let n_u and n_s be the number of eigenvalues with positive and negative real parts, respectively. These eigenvalues are associated with the corresponding *unstable* eigenspaces $V^u \in \mathbb{R}^{n_u}$ and *stable* eigenspaces $V^s \in \mathbb{R}^{n_s}$ of $e^{\mathbf{J}t}$, respectively. Trajectories in the phase portrait are moving away from the fixed point in V^u and approaching the fixed point in V^s as t increases. Since $n_u + n_s = N(M-1)$, we have the following subcases: $n_u = 0, \ldots, N(M-1)$. For $n_u = 0$, the fixed point is an attracting node and the strategy converges to this Nash point. For $n_u > 0$ and $n_u < N(M-1)$, trajectories are saddle points pointing inwards with a focus in V^s and

outwards along V^u. For $n_u = N(M-1)$, the fixed point is an $N(M-1)$-dimensional star node pointing outwards.

Due to the constraints (6.12), points on the trajectories away from the fixed point will initially reach a point on the boundary of the unit hypercube. Without loss of generality, we assume that the point is on one of the n-faces, $n \leq N(M-1)$. If the projection of the gradient is zero at that point, the trajectory will stay on the point. It is a Nash point of the game since no user can improve its payoffs by changing the strategy unilaterally. If the projected gradient is nonzero, the trajectory moves toward one of the $(n-1)$-faces of the hypercube in the direction depending on the sign of the projected gradient and reaches a point on the $(n-1)$-faces. The process will stop at any point where the projected gradient is zero or continue to move toward lower dimensional faces until the trajectory reaches one of the vertices of the hypercube $(n = 1)$. Thus, $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ converges to a Nash equilibrium or its expected rewards converge to the expected rewards of a Nash point. □

6.3.4 JRCC Algorithm

The gradient ascent algorithm in Section 6.3.3 requires the knowledge of rewards for all combinations of joint actions and the distributions of other players' actions available to each player. However, obtaining such knowledge in the JRCC game is infeasible. Due to the limitation of transmission capability, players' actions are only partially observable by other players. As a result, not all rewards can be obtained for all joint actions. More importantly, CR users and the attacker will not reveal their own action selection strategies. For these reasons, we propose the JRCC algorithm capable of selecting actions based on limited observations, updating strategy similar to gradient ascent, and obtaining the best response for each CR user individually.

The JRCC algorithm enables the cooperation between CR users with low control message overhead to facilitate CCC allocations, and adapts to PU activities and

jamming in extremely hostile environment by using the variable learning rates based on PHC and the WoLF principle [10]. When PU activity is low, the JRCC algorithm behaves like a rational hill-climbing algorithm that converges to a greedy strategy to maximize the payoffs. The performance is further improved by action strategy coordination between CR users with the exchange of a few system parameters on the established CCCs to facilitate the establishment of CCCs in future stages since their strategies for CCC selections become similar. When PU activity is high, the available CCCs under jamming attacks can be very limited, which makes the cooperation less effective. In this case, the WoLF principle can adjust the learning rates such that the players learn slowly to delay the strategy change of the opponent ("winning") or learn fast when they are outperformed by the opponent ("losing"). In addition, CR users can share local sensing decisions with neighbors to combat spectrum sensing errors and improve jamming resilience by best-effort cooperative sensing. They can also enhance jamming resilience of the network by increasing the density of CR users in the jamming area since JRCC algorithm is distributed and scalable.

The JRCC algorithm is listed in Algorithm 6.1. In each stage, each CR user selects an action that maps to a set of selected channels as CCCs for transmission, and obtains its own reward by observing the conditions of selected CCCs. (line 3–5). For action strategy coordination, each CR user broadcasts the control message with the parameters recorded in previous stage, and updates its strategy with the parameters received from neighbors (line 6–10). After the PU changes the state of the game, CR users observe the next state s' by sensing the channels (line 11). If any CCC exists, CR users exchange local sensing decisions with neighbors and make cooperative decisions individually by best-effort cooperative sensing (line 12–16). CR users then update their Q values of current state s and action a_i (line 11-17). By selecting the proper learning rate δ (line 18–22), CR users update their own strategy (line 23). The value of δ is set to the maximum for greedy strategy and a variable

Algorithm 6.1 : JRCC Algorithm for CR user $i \in \mathcal{K}$

1: *Initialize:* α, γ, ϵ, $\delta_i \in (0, 1]$, $Q(s, a) \leftarrow 0$, $\pi(s, a) \leftarrow \frac{1}{|\mathcal{A}_i|}$
2: **for** each stage t **do**
3: Select $a_i \in \mathcal{A}_i$ in state s per $\pi(s)$ with w.p. $1 - \epsilon$
4: Transmit on channels: $\{Ch : a_i \mapsto \mathcal{N}_i\}$
5: Observe U_c, N_c, P_c, J_c and calculate reward r_i
6: **if** ($U_c > 0$ and $\exists \tilde{a}_i$) **then**
7: *BroadcastToNeighbors*$(\tilde{s}, \tilde{a}_i, \tilde{\delta}_i)$
8: *ReceiveFromNeighbors*$(\tilde{s}, \tilde{a}_m, \delta_m, m \in \mathcal{K}, m \neq i)$
9: *StrategyUpdate*$(\pi(\tilde{s}, a), \tilde{a}_m, \tilde{\delta}_m)$
10: **end if**
11: Observe next state $s' \leftarrow$ *SensingChannels*$(\mathcal{N}_{s,i})$
12: **if** ($U_c > 0$) **then**
13: *BroadcastToNeighbors*(s')
14: *ReceiveFromNeighbors*$(s'_m, m \in \mathcal{K}, m \neq i)$
15: $s' \leftarrow$ *BestEffortCoopSensing*(s', s'_m)
16: **end if**
17: $Q(s, a_i) \leftarrow (1 - \alpha)Q(s, a_i) + \alpha\big[r_i + \gamma \max_b Q(s', b)\big]$
18: **if** $r_i \geq r_{th}$ **then**
19: $\delta_i = \delta_{max}$
20: **else**
21: $\delta_i =$ *WoLF*$(C(s), \pi(s, a), \bar{\pi}(s, a), Q(s, a))$
22: **end if**
23: *StrategyUpdate*$(\pi(s, a), a' = \arg\max_b Q(s, b), \delta_i)$
24: **if** ($U_c > 0$) **then** $\tilde{s} \leftarrow s, \tilde{a}_i \leftarrow a', \tilde{\delta}_i \leftarrow \delta_i$ **end if**
25: *UpdateParameters*$(\alpha, \gamma, \delta_i), s \leftarrow s'$
26: **end for**

value from the WoLF principle. The parameters $\tilde{s}$, $\tilde{a}_i$, and $\tilde{\delta}_i$ for the current greedy strategy are recorded to be broadcast in the next stage (line 24).

The strategy update and WoLF procedures are listed in Algorithm 6.2. For PHC strategy updates, the probability of the best action is increased while the probabilities of other actions are evenly decreased (line 1-9). For variable learning rates, the slow learning rate δ_w is selected for the "winning" case if the average Q value of the best action a' based on current policy π is larger than that based on average policy $\bar{\pi}$, and the fast learning rate δ_l is selected otherwise (line 10-18).

Algorithm 6.2 : Strategy Update and WoLF Procedures for JRCC Algorithm

procedure *StrategyUpdate*$(\pi(s,a), a', \delta)$
  $\delta_{sa} = \min\left(\pi(s,a), \frac{\delta}{|\mathcal{A}_i|-1}\right)$
  if $a \neq a'$ **then**
    $\Delta_{sa} = -\delta_{sa}$
  else
    $\Delta_{sa} = \sum_{a' \neq a} \delta_{sa'}$
  end if
  $\pi(s,a) \leftarrow \pi(s,a) + \Delta_{sa}$
end procedure
procedure *WoLF*$(C(s), \pi(s,a), \bar{\pi}(s,a), Q(s,a))$
  $C(s) \leftarrow C(s) + 1$
  $\bar{\pi}(s,a) \leftarrow \bar{\pi}(s,a) + \frac{1}{C(s)}\big(\pi(s,a) - \bar{\pi}(s,a)\big), \forall a' \in |\mathcal{A}_i|$
  if $\sum_{a'} \pi(s,a')Q(s,a') > \sum_{a'} \bar{\pi}(s,a')Q(s,a')$ **then**
    $\delta_i = \delta_{w_i}$
  else
    $\delta_i = \delta_{l_i}$
  end if
end procedure

6.4 Performance Analysis

In this section, we investigate JRCC performance and corresponding defense strategies under the impact of PU activities, spectrum sensing errors, CR user cooperation, and CR user deployment.

6.4.1 Effects of Primary User Activities

PU activities play the crucial role in the JRCC game as they determine the states of the game and affect the availability of the licensed channels for CR users to establish control channels. If players have the perfect observations of PU activities, both CR users and the attacker will select channels from a smaller subset of available channels as PU activities increase. As a result, the selected control channels are more susceptible to jamming attacks, as shown in Proposition 6.1. Although the attacker may initially benefit from increasing PU activities, its jamming strength is eventually constricted by high PU activities and an extremely limited number of CCCs available for jamming, as shown in Proposition 6.2.

Proposition 6.1. *Given the full and perfect observations of channel states by all players, jamming resilience ψ of CR ad hoc network $\mathcal{K}$ decreases as P_{on} increases.*

Proof. Let $C_{\mathcal{K}}$ be the number of established control links. It is a binomial distributed random variable with distribution function

$$F_{C_{\mathcal{K}}}(c) = P(C_{\mathcal{K}} \leq c) = \sum_{i=0}^{c} \binom{N_p}{i} p_c^i (1-p_c)^{N_p - i} u(c-i), \tag{6.16}$$

where $u(c)$ is the unit step function, and $p_c = P(U_c > 0)$ is the probability of successful establishment of a control link given that all control links are independent. Now we consider any control link between two CR users to be susceptible to jamming by an attacker. For simplicity, we assume that all players observe the true state s with no spectrum sensing error, and the selections of all PU-free channels are equiprobable. Let C_p be the number of PU-free channels in state s, and $C_p \leq N_s$ for PU activity-limited scenarios, and recall that U_c, N_c, and J_c are the number of valid control channels, the selected common channels, and the jammed common channels, respectively. The probability of successfully establishing the control link is

$$p_c = P(U_c > 0) = \sum_{m=1}^{C_p} P(N_c = m) P(J_c \leq m-1 | N_c = m), \tag{6.17}$$

where $P(N_c = m)$ is the probability of CR users selecting m common channels for the control link given by

$$P(N_c = m) = \frac{\binom{C_p}{m} \sum_{i=0}^{C_p - m} \binom{C_p - m}{i} \sum_{j=0}^{C_p - m - i} \binom{C_p - m - i}{j}}{\left(\sum_{i=0}^{C_p} \binom{C_p}{i} \right)^2} \tag{6.18}$$

and $P(J_c \leq m-1 | N_c = m)$ is the probability of the attacker jamming at most $m-1$ out of those m channels selected by CR users given by

$$P(J_c \leq m-1 | N_c = m) = \frac{\sum_{i=0}^{m-1} \binom{m}{i} \sum_{j=0}^{C_p - m} \binom{C_p - m}{j}}{\sum_{i=0}^{C_p} \binom{C_p}{i}}. \tag{6.19}$$

The first binomial coefficient $\binom{C_p}{m}$ in the numerator of (6.18) is the number of choices of one CR user selecting m out of total C_p channels. The second $\binom{C_p - m}{i}$ is the

combinations of selecting i channels out of the rest of $C_p - m$ channels for the same CR user, in addition to those m common channels. The third $\binom{C_p-m-i}{j}$ is the number of the other CR user's choices of selecting j channels from the remaining $C_p - m - i$ channels not selected by the first CR user in addition to those m common channels. The first binomial coefficient $\binom{m}{i}$ in the numerator of (6.19) says which i out of those m common channels are jammed, and the second $\binom{C_p-m}{j}$ is the number of attacker's choices of selecting j out of the rest of $C_p - m$ channels in addition to those i jammed channels. Since the average number of available channels is $E[C_p] = \sum_{i=1}^{N_p}(1 - P_{\text{on}}^i) = (1 - P_{\text{on}})N_p$, as P_{on} increases, the average number of C_p decreases. Thus, from (6.2) to (6.19), we know that p_c decreases as P_{on} increases. From (6.16), we have decreasing $F_{C_\mathcal{K}}(c)$ as p_c decreases given c and N_p. Thus, $E[C_\mathcal{K}] = N_p p_c$ decreases accordingly and, by using (6.3) with fixed $L_\mathcal{K}$, the result follows. □

Proposition 6.2. *Given the full and perfect observations of channel states by all players, jamming strength ζ of the attacker increases for $P_{on} \leq p_\zeta$ and decreases for $P_{on} > p_\zeta$, $p_\zeta \in (0, 1)$ as P_{on} increases.*

Proof. Based on the same assumptions in the proof of Proposition 6.1, let $D_\mathcal{K}$ be a binomial distributed random variable counting the number of jammed control links with distribution function

$$F_{D_\mathcal{K}}(d) = P(D_\mathcal{K} \leq d) = \sum_{i=0}^{d} \binom{N_p}{i} p_j^i (1 - p_j)^{N_p - i} u(d - i), \tag{6.20}$$

where $u(c)$ is the unit step function, and p_j is the probability of successfully jamming a control link given by

$$p_j = \sum_{m=1}^{C_p} P(J_c = m | N_c = m) P(N_c = m), \tag{6.21}$$

where $P(N_c = m)$ is from (6.18) and $P(J_c = m|N_c = m)$ is given by

$$P(J_c = m|N_c = m) = \frac{\sum_{j=0}^{C_p-m} \binom{C_p-m}{j}}{\sum_{i=0}^{C_p} \binom{C_p}{i}}. \tag{6.22}$$

The binomial coefficient $\binom{C_p-m}{j}$ is simply the number of attacker's choices of selecting j out of the rest of C_p-m channels in addition to m jammed channels. We know that $P(J_c = m|N_c = m)$ in (6.22) increases as m decreases. As C_p decreases initially due to increasing P_{on}, the combinations of channel selections are reduced such that the probability of successful jamming p_j in (6.21) increases. When C_p is small due to high P_{on} that limits $P(N_c = m)$ in (6.18) and jamming opportunities $P(J_c = m|N_c = m)$ in (6.22), p_j decreases. $P_{\text{on}} = p_\zeta$ is the point where p_j reaches the maximum value. Thus, p_j increases for low and moderate values of P_{on}, reaches its maximum value when $P_{\text{on}} = p_\zeta$, and reduces to zero as P_{on} approaches one. From $E[D_\mathcal{K}] = N_p p_j$ and (6.4) with fixed $L_\mathcal{K}$, the result follows. □

6.4.2 Effects of Spectrum Sensing Errors

In addition to the effects of PU activities, spectrum sensing errors can have significant impact on control channel allocation under jamming attacks. This is because spectrum sensing determines whether or not the true state of PU activities or channel availability is observed. Spectrum sensing determines the presence of PUs based on binary hypothesis testing: the null hypothesis H_0 for the absence of PUs and the alternative hypothesis H_1 for the presence of PUs. Its performance is evaluated in terms of probability of false alarm P_f in H_0 and probability of miss detection P_m in H_1. The effects of these two types of sensing errors are discussed as follows.

6.4.2.1 Impact of False Alarms

A false alarm is a type I error that CR users or attackers falsely detect the presence of PUs when the channel is vacant (determining H_1 given H_0). In this case, CR users are mistakenly forced to select CCCs from a much smaller subset of available channels for allocations, which incurs a higher risk of jamming. In the extreme cases, they may select channels from exclusive subsets of channels failing to establish any control link. Moreover, CR user cooperation can be considerably compromised by

false alarm errors because a single CR user making a false alarm error can result in different states observed by two CR users and the failure to establish the control link. Therefore, the impact of false alarm errors on jamming resilience can be quite significant even if P_f is small. This is shown in Proposition 6.3.

Proposition 6.3. *Given $P_m = 0$ and fixed P_{on}, the jamming resilience ψ of CR ad hoc network $\mathcal{K}$ decreases as P_f increases.*

Proof. We consider the control link between two CR users under the jamming of an attacker in state s. Let $\mathcal{S}_0$ and $\mathcal{S}_0^c$ be the set of channels unoccupied and occupied by PUs, respectively, in true state s, and $\mathcal{S}_0 \cup \mathcal{S}_0^c = \mathcal{N}_p$. Let $\mathcal{N}_n$ and $\mathcal{N}_n^c$ be the set of channels observed by player n to be unoccupied and occupied by PUs, respectively, as the result of making false alarm errors with probability P_f in each channel. Note that $\mathcal{N}_n^c$ includes PU-free channels due to false alarm errors, and $\mathcal{N}_n = \mathcal{S}_0 - \mathcal{F}_n$, where $\mathcal{F}_n$ is the set of PU-free channels with false alarm errors made by player n. Here player n is either CR user k, $k \in \{1, 2\}$, or attacker j.

Let $\mathcal{N}_{s,n}$ be the set of channels selected by player n and $\mathcal{U}_c$ be the set of valid CCCs. If we consider any PU-free channel $c \in \mathcal{S}_0$, the probability of c being a valid CCC $c_v \in \mathcal{U}_c$ is

$$
\begin{aligned}
p_c &= P(c \in \mathcal{U}_c \mid c \in \mathcal{S}_0) = P(c \in \mathcal{U}_c) && (6.23)\\
&= \Big[\prod_k P(c \in \mathcal{N}_{s,k})\Big] P(c \notin \mathcal{N}_{s,j}) && (6.24)\\
&= \Big[\prod_k P(c \in \mathcal{N}_k)\Big] P(c \notin \mathcal{N}_j) && (6.25)\\
&= \Big[\prod_k P(c \in \{\mathcal{S}_0 - \mathcal{F}_k\})\Big] P(c \notin \{\mathcal{S}_0 - \mathcal{F}_j\}) && (6.26)\\
&= P\big(c \in \{\mathcal{S}_0 - \{\cup_k \mathcal{F}_k\}\}\big) P(c \in \mathcal{F}_j) && (6.27)\\
&= \big(1 - P(c \in \{\cup_k \mathcal{F}_k\})\big) P(c \in \mathcal{F}_j) && (6.28)
\end{aligned}
$$

The equality in (6.24) follows that c_v is a channel selected by both CR users but not selected by the attacker. Note that the channels in $\mathcal{N}_{s,k}$ do not have false alarm

errors because CR users consider channels with false alarm errors occupied by PUs. The equality in (6.25) follows $\mathcal{N}_{s,n} \subseteq \mathcal{N}_n$ and the assumption that all players select all available channels for maximizing the probability of establishing or jamming the control link. When P_f increases, $P(c \in \mathcal{F}_n)$ also increases. As a result, $P(c \in \{\cup_k \mathcal{F}_k\})$ decreases while $P(c \in \mathcal{F}_j)$ increases in (6.28). However, the decrease in $P(c \in \{\cup_k \mathcal{F}_k\})$ caused by false alarms of at least one CR user is greater than the increase in $P(c \in \mathcal{F}_j)$ caused by false alarms of the attacker. Thus, we have $P(c \in \mathcal{U}_c)$ decreases as P_f increases, indicating that the probability of establishing this control link p_c decreases as P_f increases. Since this result can be applied to any state s and any link between two CR users, we conclude from (6.16) and (6.28) that the average number of successful control links $E[C_{\mathcal{K}}] = N_p p_c$ decreases as P_f increases. Using (6.3) with fixed $L_{\mathcal{K}}$, the result follows. □

6.4.2.2 Impact of Miss Detection

Miss detection is a type II error that CR users or attackers mistakenly detect no presence of PUs when the PU occupies the channel (determining H_0 given H_1). Unlike false alarms, miss detection affects only those PU-occupied channels that are unavailable for CR users to establish control links. Thus, the effect of miss detection on CCC establishment is relatively minor compared to those of false alarms. However, miss detection still causes performance degradation because channel selections based on the observed states may include PU-occupied channels that cannot be used for CCCs. For the attacker, the effect of miss detection is more severe than those of false alarms. This is because, with limited transmission capability and miss detection errors, the attacker is more likely to select PU occupied channels to jam. Therefore, jamming is less effective for higher P_m. This is shown in Proposition 6.4.

Proposition 6.4. *Given $P_f = 0$ and fixed P_{on}, jamming strength ζ of the attacker decreases as P_m increases.*

Proof. We consider the control link between two CR users under the jamming of an attacker in state s. $\mathcal{S}_0$ and $\mathcal{S}_0^c$ are defined as before. Let $\mathcal{N}_n$ and $\mathcal{N}_n^c$ be the set of channels observed by player n to be unoccupied and occupied by PUs, respectively, as the result of making miss detection errors with probability P_m in each channel. Note that $\mathcal{N}_n$ includes PU-occupied channels due to miss detection errors, and $\mathcal{N}_n = \mathcal{S}_0 + \mathcal{M}_n$, where $\mathcal{M}_n$ is the set of PU-occupied channels with miss detection errors made by player n. Here player n is either CR user k, $k \in \{1, 2\}$, or attacker j.

Let $\mathcal{N}_{s,n}$ be the set of channels selected by player n, and $\mathcal{J}_c$ be the set of jammed CCCs. For successful jamming, a channel c_n selected by player n cannot be the one with the miss detection error. Since player n independently selects channels observed to be PU-free in $\mathcal{N}_n$, the probability of successfully jamming the control link is

$$
\begin{aligned}
p_j &= P(c_1, c_2, c_j \in \mathcal{J}_c \mid c_1 \in \mathcal{N}_1, c_2 \in \mathcal{N}_2, c_j \in \mathcal{N}_j) && (6.29)\\
&= \prod_n P(c_n \in \mathcal{J}_c \mid c_n \in \mathcal{N}_n) && (6.30)\\
&= \prod_n \frac{P(c_n \in \mathcal{J}_c)}{P(c_n \in \mathcal{N}_n)} && (6.31)\\
&= \prod_n \frac{P(c_n \in \mathcal{J}_c)}{P(c_n \in \{\mathcal{S}_0 + \mathcal{M}_n\})} && (6.32)\\
&= \prod_n \frac{P(c_n \in \mathcal{J}_c)}{P(c_n \in \mathcal{S}_0) + P(c_n \in \mathcal{M}_n)} && (6.33)
\end{aligned}
$$

As P_m increases, the denominator of (6.33) increases due to the increase in $P(c_n \in \mathcal{M}_n)$. Moreover, $P(c_n \in \mathcal{J}_c)$ in the numerator of (6.33) decreases as P_m increases. This is because miss detection errors of any player reduce the number of valid CCCs established by CR users and jammed by the attacker. Thus, from (6.33), p_j decreases as P_m increases. The analysis can be applied to any state and any control link between two CR users in the jamming region. From (6.20) and (6.33), we conclude that the average number of jammed control links $E[D_{\mathcal{K}}] = N_p p_j$ decreases as P_m increases. Using (6.4) with fixed $L_{\mathcal{K}}$, the result follows. □

6.5 Intrusion Defense Strategies

In this section, we analyze proposed intrusion defense strategies against intelligent jamming attacks. These countermeasures are action strategy coordination, best-effort cooperative sensing, and CR user deployment density and scalability as follows.

6.5.1 Action Strategy Coordination

Action strategy coordination (ASC) is a defense mechanism for neighboring CR users to facilitate the establishment of control channels in future stages and enhance jamming resilience by coordinating their action strategies on established control channels. The coordination among users is achieved by the exchange of a short control message that includes coordination parameters: current state s, selected action in this state $a(s)$, and learning rate δ. CR users update their action selection strategy with their own coordination parameters and those received from their neighbors. Such strategy updates increase the levels of similarity in action selection strategies of neighboring CR users, which increases the probability of selecting commonly available channels as control channels. We first define the similarity of two strategies in Definition 6.6 by extending the definition in [31] to comparing strategies of different sizes, and then show that action strategy coordination increases the similarity of CR users' strategies in Theorem 6.2.

Definition 6.6 (Similarity of Action Strategies)**.** *The similarity of two action strategies: $\pi_1(s_1)$ of state s_1 and $\pi_2(s_2)$ of state s_2, is defined as*

$$\xi(\pi_1, \pi_2) = - \parallel f(\pi_1) - f(\pi_2) \parallel^2, \tag{6.34}$$

where $f : \mathbb{R}^{|\mathcal{A}(s_i)|} \rightarrow \mathbb{R}^{|\mathcal{A}(s_0)|}$ is a mapping function that maps action space $\mathcal{A}(s_i)$ to $\mathcal{A}(s_0)$, $\mathcal{A}(s_i) \subseteq \mathcal{A}(s_0)$, $\forall i$, by setting $\pi_{i,m} = 0$ for $a_{i,m} \in \mathcal{A}(s_0)$ and $a_{i,m} \notin \mathcal{A}(s_i)$, and s_0 is the state with no PU activity in all licensed channels. As a special case, $\xi(\pi_1, \pi_2) = - \parallel \pi_1 - \pi_2 \parallel^2$ if $s_1 = s_2$.

Theorem 6.2 (Action Strategy Coordination). *Let $\pi_i(s) = [x_{ij}]$, $j = 1, \ldots, |\mathcal{A}_i(s)|$, be the action strategy of CR user i, where x_{ij} is the probability of selecting action $a_{i,j}$ in state s by following $\pi_i(s)$. Given coordination parameters $\Delta_i = (s, a_i, \delta_i)$ for strategy updates at CR user i, action strategy coordination of JRCC algorithm achieves higher similarity ξ than strategy updates without coordination.*

Proof. Given two strategies: $\pi_1^t(s) = [x_{11}, \ldots, x_{1m}]$ and $\pi_2^t(s) = [x_{21}, \ldots, x_{2m}]$ of stage t where $m = |\mathcal{A}(s)|$, we can first find the similarity between $\pi_1(s)$ and $\pi_2(s)$ before strategy updates as

$$\xi_{\text{orig}} = \xi(\pi_1^t, \pi_2^t) = -\sum_j (x_{2j} - x_{1j})^2. \tag{6.35}$$

Without loss of generality, we let $a_1 = a_{1k}$ and $a_2 = a_{2n}$, $k, n \in \{1, \ldots, |\mathcal{A}_i(s)|\}$. Without strategy coordination, CR user i updates its strategy $\pi_i^t(s)$ with Δ_i. According to the strategy update procedure in Algorithm 6.1, we have updated strategies as $\pi_1^{t+1}(s) = [x_{11} - \frac{\delta_1}{m-1}, \ldots, x_{1k} + \delta_1, \ldots, x_{1m} - \frac{\delta_1}{m-1}]$ and $\pi_2^{t+1}(s) = [x_{21} - \frac{\delta_2}{m-1}, \ldots, x_{1n} + \delta_2, \ldots, x_{2m} - \frac{\delta_2}{m-1}]$. The similarity of $\pi_1^{t+1}(s)$ and $\pi_2^{t+1}(s)$ without coordination is

$$\begin{aligned} \xi_{\text{indv}} &= \xi(\pi_1^{t+1}, \pi_2^{t+1}) && (6.36) \\ &= -\sum_{j \neq k,n} \left(x_{2j} - x_{1j} + \frac{\delta_1 - \delta_2}{m-1}\right)^2 \\ &\quad -\left(x_{1k} - x_{2k} + \delta_1 + \frac{\delta_2}{m-1}\right)^2 - \left(x_{2n} - x_{1n} + \delta_2 + \frac{\delta_1}{m-1}\right)^2 && (6.37) \end{aligned}$$

Since the term $\frac{\delta_1 - \delta_2}{m-1}$, $(\delta_1 + \frac{\delta_2}{m-1})$, and $(\delta_2 + \frac{\delta_1}{m-1})$ in (6.37) are generally nonzero, we have $\xi(\pi_1^{t+1}, \pi_2^{t+1}) < \xi(\pi_1^t, \pi_2^t)$. The strategies of two CR users become less similar after strategy updates without coordination. If CR users coordinate strategy updates, each CR user updates its strategy with both Δ_1 and Δ_2 in arbitrary order. We then have updated strategies as $\pi_i^{t+1}(s) = [x_{i1} - \frac{\delta_1 + \delta_2}{m-1}, \ldots, x_{ik} + \delta_1 - \frac{\delta_2}{m-1}, \ldots, x_{in} + \delta_2 - \frac{\delta_1}{m-1}, \ldots, x_{im} - \frac{\delta_1 + \delta_2}{m-1}]$, $i = 1, 2$. We can easily find the similarity between these two strategies as

$$\xi_{\text{jrcc}} = \xi(\pi_1^{t+1}, \pi_2^{t+1}) = -\sum_j (x_{2j} - x_{1j})^2 = \xi(\pi_1^t, \pi_2^t) = \xi_{\text{orig}}. \tag{6.38}$$

Therefore, we have $\xi_{\text{indv}} < \xi_{\text{jrcc}}$. That is, ASC maintains the same level of similarity after the updates, which is higher than ξ values obtained by the updates without ASC. The proof can be easily generalized to more than two CR users, which is omitted here due to the lack of space. □

In addition to achieving the similarity, ASC increases the probability of selecting same actions, which generally leads to higher jamming resilience. In the proof of Theorem 6.2, we know that, after strategy updates with no ASC, the probability of selecting a_{1k} and a_{2n} is the largest in the strategy of CR user 1 and 2, respectively. Thus, CR users with no ASC tend to choose different actions that may map to completely different subsets of channels leading to no common channels for valid CCC links. For updates with ASC, the probabilities of selecting a_{ik} and a_{in} are the largest among all actions and the probability of choosing either action by both CR users is comparable. Thus, the probability of choosing the same actions is higher with ASC, leading to higher probability of selecting common channels. This is shown in Corollary 6.1:

Corollary 6.1. *Let p_{orig}, p_{indv}, p_{jrcc} be the probability of selecting same actions before strategy updates, after updates without ASC, and after updates with ASC, respectively. Given $\pi_i(s)$ and $\Delta_i = (s, a_i, \delta_i)$ of CR user i, $i \in \{1, 2\}$, for $m = |\mathcal{A}(s)| \gg 1$, ASC achieves higher probability as*

$$p_{\text{jrcc}} = p_{\text{orig}} + \sum_i (\delta_1 x_{ik} + \delta_2 x_{in} + \delta_i^2) > p_{\text{orig}} \tag{6.39}$$

$$= p_{\text{indv}} + \delta_1(x_{1k} + \delta_1) + \delta_2(x_{2n} + \delta_2) > p_{\text{indv}} \tag{6.40}$$

where x_{ik} and x_{in} are the probability of CR user i selecting action a_{ik} and a_{in}, respectively.

Proof. We continue from the proof of Theorem 6.2 and find the probability of selecting

same actions before updates as

$$p_{\text{orig}} = \sum_{j} \prod_{i} x_{ij}. \tag{6.41}$$

Using $\pi_i^{t+1}(s)$ in the previous proof, we obtain the probability of choosing same actions after individual strategy updates as

$$\begin{aligned} p_{\text{indv}} &= \Big(\sum_{j \neq k,n} \prod_{i} \Big(x_{ij} - \frac{\delta_i}{m-1} \Big) \Big) \\ &+ (x_{1k} + \delta_1) \Big(x_{2k} - \frac{\delta_2}{m-1} \Big) + (x_{2n} + \delta_2) \Big(x_{1n} - \frac{\delta_1}{m-1} \Big) \end{aligned} \tag{6.42}$$

Similarly, we obtain the probability of selecting same actions after ASC as

$$\begin{aligned} p_{\text{jrcc}} &= \Big(\sum_{j \neq k,n} \prod_{i} \Big(x_{ij} - \frac{\delta_1 + \delta_2}{m-1} \Big) \Big) \\ &+ \prod_{i} \Big(x_{ik} + \delta_1 - \frac{\delta_2}{m-1} \Big) + \prod_{i} \Big(x_{in} + \delta_2 - \frac{\delta_1}{m-1} \Big) \end{aligned} \tag{6.43}$$

Since $m \gg 1$ and $0 < \delta_i \leq 1$, we can approximate (6.42) and (6.43) by removing $\frac{\delta_i}{m-1}$ terms. After some algebra manipulation, we obtain (6.39) and (6.40) by comparing (6.43) with (6.41) and (6.42), respectively. □

6.5.2 Best-Effort Cooperative Sensing

In Section 6.4.2, we show the impact of spectrum sensing errors on jamming resilience. To mitigate such an impact, especially the impact of false alarms, we propose best-effort cooperative sensing (BCS) as the countermeasure to combat jamming attacks and enhance jamming resilience by reducing sensing errors. BCS is a distributed cooperative sensing scheme that CR users make the best efforts to share local sensing data with neighbors by using control links established in the previous stage yet still valid in the current stage, and individually make sensing decisions based on any collected sensing data. Unlike conventional distributed sensing schemes, BCS does not require the participation of all neighbors or an unanimous decision reached by multiple iterations of message exchanges. This is because control links between neighbors are

not guaranteed under PU activities and jamming. Therefore, the number of neighbors participating in BCS is a random variable that determines the achievable cooperative probabilities of false alarm Q_f and miss detection Q_m.

Let B_k be the number of CR users participating in BCS at CR user k. When $B_k = 1$, BCS degenerates to local spectrum sensing at CR user k with $Q_f(1) = P_f$ and $Q_m(1) = P_m$. In conventional cooperative sensing, the most popular "K out of N" data fusion rules for hard combinations of local sensing decisions are AND, OR, and majority rules [3]. It is known that AND (OR) rule significantly reduces Q_f (Q_m) at the cost of increasing Q_m (Q_f). The majority rule achieves balanced performance in both Q_f and Q_m. Thus, we show that BCS with majority rule achieves better $Q_f(B_k)$ performance with the cooperation of arbitrary number of neighbors than local spectrum sensing in Proposition 6.5.

Proposition 6.5. *For $P_f, P_m < 0.5$ and an odd integer $B_k > 2$, best-effort cooperative sensing with the majority rule improves $Q_f(B_k)$ from P_f and $Q_m(B_k)$ from P_m by satisfying $Q_f(B_k+1), Q_f(B_k+2) < Q_f(B_k) < P_f$ and $Q_m(B_k+1), Q_m(B_k+2) < Q_m(B_k) < P_m$, respectively.*

Proof. We first focus on Q_f. Using (5.35) with same P_f for all CR users and no reporting errors $P_e = 0$, we have

$$Q_f(B_k) = \sum_{\ell=\lfloor \frac{B_k}{2} \rfloor + 1}^{B_k} \binom{B_k}{\ell} P_f^{\ell} (1-P_f)^{B_k - \ell}. \tag{6.44}$$

$B_k = 2$ is the special case: $Q_f(2) = (P_f)^2 < Q_f(1) = P_f$. We prove this by mathematical induction in pairs of B_k for $B_k > 2$ due to the floor function for choosing ℓ in (6.44). When $B_k = 3$, $Q_f(3) = 3(P_f)^2(1-P_f) + (P_f)^3$. We can also find $Q_f(4) = 4(P_f)^3(1-P_f) + (P_f)^4$ and $Q_f(5) = 10(P_f)^3(1-P_f)^2 + 5(P_f)^4(1-P_f) + (P_f)^5$, respectively. It is evident that $Q_f(4), Q_f(5) < Q_f(3) < P_f$ for $P_f < 0.5$. If $Q_f(n-1), Q_f(n) < Q_f(n-2) < P_f$ is valid for an odd n, we want to show that

$Q_f(n+1), Q_f(n+2) < Q_f(n) < P_f$. Setting $B_k = n+1$ in (6.44), we obtain

$$\begin{aligned} Q_f(n+1) &= \sum_{\ell=\frac{n+3}{2}}^{n+1} \binom{n+1}{\ell} (P_f)^{\ell}(1-P_f)^{n-\ell+1} && (6.45) \\ &= \sum_{\ell'=\frac{n+1}{2}}^{n} \binom{n+1}{\ell'+1} (P_f)^{\ell'+1}(1-P_f)^{n-\ell'+1} && (6.46) \\ &= \sum_{\ell'=\frac{n+1}{2}}^{n} \left[P_f \binom{n+1}{\ell'+1} \right] (P_f)^{\ell'}(1-P_f)^{n-\ell'} && (6.47) \end{aligned}$$

The second equality in (6.46) is obtained by change of variables ($\ell = \ell+1$). The third equality in (6.47) is obtained by moving P_f inside the brackets as part of binomial coefficients. Since $Q_f(n+1)$ and $Q_f(n) = \sum_{k=\frac{n+1}{2}}^{n} \binom{n}{k} (P_f)^{\ell}(1-P_f)^{n-k}$ have the same number of terms in summation, we can compare their binomial coefficients term by term as

$$\begin{aligned} P_f \binom{n+1}{\ell'+1} &= P_f \left(\frac{n+1}{\ell'+1} \right) \binom{n}{\ell'} && (6.48) \\ &\leq 2P_f \left(\frac{n+1}{n+3} \right) \binom{n}{\ell'} \leq \binom{n}{\ell'} = \binom{n}{k} && (6.49) \end{aligned}$$

The first inequality holds for $\ell' \geq \frac{n+1}{2}$. The second inequality holds for $2P_f(\frac{n+1}{n+3}) < 1$ and $P_f < 0.5$. Using $Q(n) < Q_f(n-2) < P_f$, we conclude that $Q(n+1) < Q(n) < P_f$. Similarly, by setting $B_k = n+2$ in (6.44), we have

$$\begin{aligned} Q_f(n+2) &= \sum_{\ell=\frac{n+3}{2}}^{n+2} \binom{n+2}{\ell} (P_f)^{\ell}(1-P_f)^{n-\ell+2} && (6.50) \\ &= \sum_{\ell'=\frac{n+1}{2}}^{n} \binom{n+2}{\ell'+1} (P_f)^{\ell'+1}(1-P_f)^{n-\ell'+1} + (P_f)^{n+2} && (6.51) \\ &\approx \sum_{\ell'=\frac{n+1}{2}}^{n} \left[P_f(1-P_f) \binom{n+2}{\ell'+1} \right] (P_f)^{\ell'}(1-P_f)^{n-\ell'} && (6.52) \end{aligned}$$

The second equality in (6.51) is obtained by change of variables ($\ell = \ell'+1$) and separating the last term $(P_f)^{n+2}$ from the summation. The approximation in (6.52) is obtained by moving $P_f(1-P_f)$ inside the brackets and dropping the last term,

which is very small ($P_f^{n+2} \approx 0$). We again compare the coefficients of $Q_f(n+2)$ and $Q_f(n)$ term by term as

$$\begin{aligned} P_f(1-P_f)\binom{n+2}{\ell'+1} &= \frac{P_f(1-P_f)(n+2)(n+1)}{(\ell'+1)(n-\ell'+1)}\binom{n}{\ell'} && (6.53) \\ &\leq 4P_f(1-P_f)\Big(\frac{n+2}{n+3}\Big)\binom{n}{\ell'} && (6.54) \\ &\leq \binom{n}{\ell'} = \binom{n}{k} && (6.55) \end{aligned}$$

The first inequality holds for $\ell' \geq \frac{n+1}{2}$. The second inequality holds because $4P_f(1-P_f)(\frac{n+2}{n+3}) < 1$ for $P_f(1-P_f) < 0.25$. Therefore, by using $Q(n) < Q_f(n-2) < P_f$, we conclude that $Q(n+1), Q(n+2) < Q_f(n) < P_f$.

To find Q_m, we replace Q_f and P_f with Q_d and P_d, respectively, in (6.44), where $Q_d = 1-Q_m$ and $P_d = 1-P_m$ are cooperative and individual probability of detection, respectively. By using the same techniques, we can show that $Q_m(B_k+1), Q_m(B_k+2) < Q_m(B_k) < P_m$ for $P_m < 0.5$ and any odd integer $B_k > 2$. □

Based on Proposition 6.5, we show in Theorem 6.3 that jamming resilience is enhanced by BCS with hard combinations and the majority rule.

Theorem 6.3 (Best-Effort Cooperative Sensing)**.** *Best-effort cooperative sensing improves jamming resilience ψ of CR ad hoc network $\mathcal{K}$.*

Proof. The result follows Proposition 6.5 that BCS decreases the probability of false alarm Q_f, and Proposition 6.3 that jamming resilience ψ of CR ad hoc network $\mathcal{K}$ increases as Q_f decreases. □

6.5.3 Deployment Density and Scalability

The deployment of CR users in a given area, that is, deployment density, is a deciding factor of the reliability and connectivity in CR ad hoc networks [54]. As a result, increasing CR user density in the jamming region can be utilized as a defense strategy to provide resilience to jamming. The main reason that jamming mitigation can

be achieved is because, as deployment density D_K increases, the average distance between CR users decreases while the average distance between the attacker and CR users remains the same. This results in better SINR and less effective jamming perceived at CR users. This is shown in Theorem 6.4.

Theorem 6.4 (Deployment Density)**.** *Given the jamming region $\mathfrak{A}$, jamming resilience ψ increases as the density of uniformly distributed CR users D_K increases in $\mathfrak{A}$.*

Proof. We look at the effects of the increase of D_K on the interference and signal power observed at CR users. Let R_j be the radius of the jamming region $\mathfrak{A}$ with the attacker at the center of the circle and D be the distance between any uniformly distributed CR user inside $\mathfrak{A}$ and the attacker at the center of $\mathfrak{A}$. From [68], we know that the distribution of D is $f_D(d) = \frac{2d}{R_j^2}$. Hence, the average distance $E[D] = \bar{D} = \frac{2}{3}R_j$ is independent of D_K or the number of CR users in $\mathfrak{A}$. Let $\mathfrak{D}^+$ and $\mathfrak{D}^-$ be the area outside and inside, respectively, the circle of radius $\bar{D}$ in $\mathfrak{A}$ and $\mathfrak{D}^+ \cup \mathfrak{D}^- = \mathfrak{A}$. We can find the number of CR users in $\mathfrak{D}^+$ to be $K_{\mathfrak{D}^+} = \frac{5}{9}K$. As K increases, the increase in $K_{\mathfrak{D}^+}$ is more than that in $K_{\mathfrak{D}^-}$. As a result, the average received interference power at CR users decreases as K increases.

Let d_k be the distance to the k-th nearest neighbor of any CR user i in $\mathfrak{A}$. From

$$D_K = \frac{K}{\pi R_j^2} = \frac{k}{\pi d_k^2}, \tag{6.56}$$

we obtain $d_k = \frac{kR_j}{\sqrt{K}} \propto K^{-\frac{1}{2}}$. The received signal power P_{ik} at the CR user i from its neighbor k is then $P_{ik} \propto d_k^{-\eta} \propto K^{\frac{\eta}{2}}$ for path loss exponent $\eta \geq 2$. As K increases, CR users receive higher signal power from neighbors. This results in higher SINR and fewer CCC links being successfully jammed. Thus, jamming resilience ψ increases. □

6.6 Performance Evaluation

In this section, we validate the analysis in Section 6.4 and evaluate the performance of the proposed JRCC algorithm. We first show the convergence of JRCC algorithm, and then show how JRCC effectively establishes control channels and maintains network connectivity under intelligent jamming attacks, and improves jamming resilience by using the proposed intrusion defense countermeasures such as action strategy coordination, distributed cooperative sensing, and increase of deployment density.

In the simulation environment, we set $N = 4$ ($K = 3$, $J = 1$), $N_p = 6$, and $N_s = 3$ unless otherwise specified. CR users are uniformly distributed in the jamming region $\mathfrak{A}$, a circular area with the attacker at the center and radius 50 m. PU activities $P_{\text{on}}/P_{\text{off}}$ are the same in every channel. For channel model parameters, we set transmit power to 100 mW, $\eta = 3.3$ for path loss, SINR threshold $\gamma_{th} = 4$ dB, channel bandwidth 100 kHz in 900 MHz band. The noise floor is set to -101 dBm. For multiagent reinforcement learning, we set $\alpha^t = 1/(1 + t/500), \delta_w^t = 1/(1 + t/10)$ where t is stage index, $\delta_l = 4\delta_w$, $\gamma = 0.9$, $\epsilon = 0.1$, $\delta_{max} = 1$, and $r_{th} = 0.5$. The simulation results are averaged over 50 runs.

6.6.1 Convergence of JRCC Algorithm

Figure 6.2 plots the average rewards of 9 CR users and the attacker with P_{on} set to 0.3 and perfect spectrum sensing ($P_f = P_m = 0$) in an exemplary run. The figure clearly shows the convergence of JRCC algorithm and validates Theorem 6.1 that the rewards of players converge to the rewards of a Nash equilibrium.

6.6.2 Transmission Capability

Figure 6.3 shows jamming resilience ψ or jamming strength ζ of players for different number of N_s given $N_p = 6$ and $P_{\text{on}} = 0$ and 0.5. In capability-limited cases (no PU activity), ψ increases as N_s increases from 1 to 4, but the increases in ψ are monotonically decreasing and eventually turning into the decreases in ψ from 4 to 6.

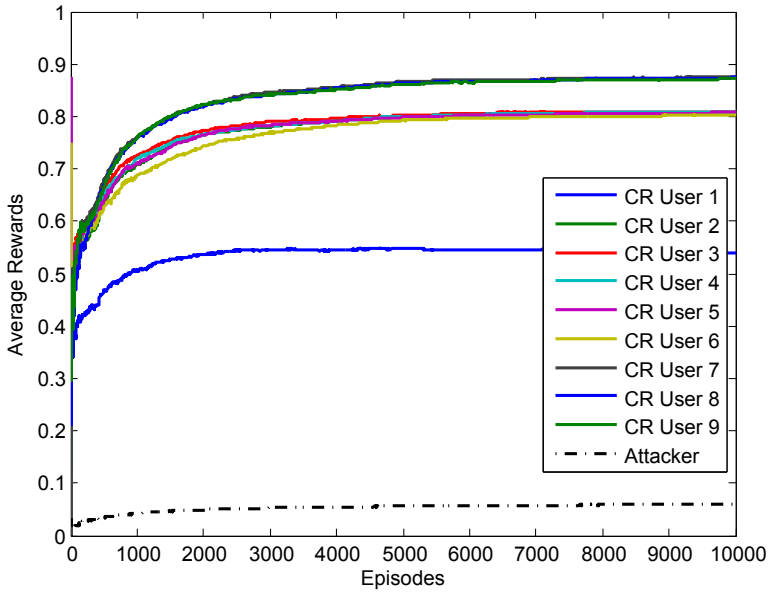


Figure 6.2: Convergence of JRCC Algorithm.

ζ, on the other hand, exhibits larger values when N_s is either small or large ($N_s = 1$ or 6 in this case). When N_s is small, the combinations of channel selections are small, which makes the jamming easier for the attacker. When N_s is large, the possibility of selecting more channels by CR users increases the percentage of jammed control links because the attacker can match the transmission capability of CR users. Thus, the optimal value of N_s for CR users is in the middle ($N_s = 4$ in this case). This shows that transmitting on all channels is not necessarily the best strategy for CR users in capability-limited cases if the attacker has the same capability. As P_{on} increases from 0, the capability-limited cases gradually become activity-limited scenarios where channel availability and selections are limited more by PU activities rather than by transmission capability. For example, the differences between values of ψ or ζ for large values of N_s ($N_s > 3$) are very small for $P_{\text{on}} = 0.5$ because the average number of channels available for selection is around 3. Therefore, in this case, increasing N_s beyond 3 does not improve ψ for CR users or ζ for the attacker.

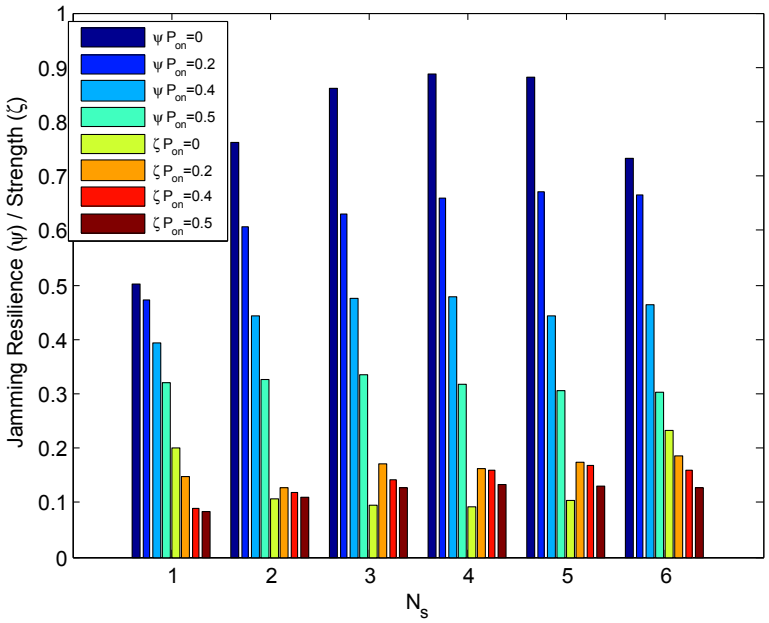


Figure 6.3: Jamming Resilience and Jamming Strength versus Transmission Capability N_s.

6.6.3 Action Strategy Coordination

PU activity is one of the major crucial factors of JRCC performance since it mainly determines the available channels for CCC allocations. Figure 6.4 shows jamming resilience ψ of CR users and jamming strength ζ of the attacker under different degrees of PU activities. Note that, for avoiding the overlapping of curves and improving clarity, inverted strength $\phi = 1 - \zeta$ is shown for attacker's jamming strength. In the figure, JRCC is compared to Random, PHC, and WoLF-PHC algorithms. Here "Random" is a scheme with random action selections with no learning. PHC is a greedy algorithm that improves the policy by selecting actions according to maximum Q values. WoLF-PHC is based on PHC with variable learning rates determined by the WoLF principle [10]. Both CR users and the attacker use the same algorithm in the game except that, in the case of JRCC, action strategy coordination is not available for the single attacker.

We first observe that jamming resilience decreases as P_{on} for all methods. The

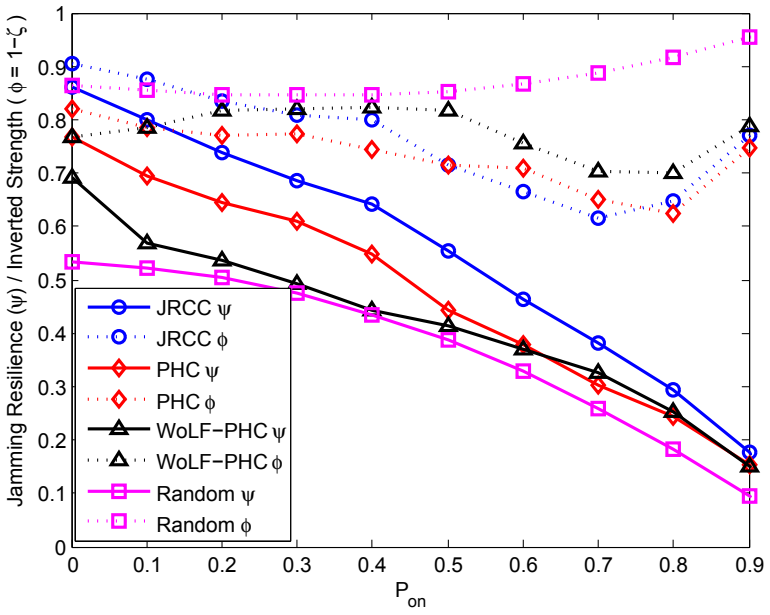


Figure 6.4: Jamming Resilience and Jamming Strength vs. PU Activities (P_{on}) with Perfect Sensing.

"Random" has both the lowest ψ and the lowest ζ among all methods because it maximizes the uncertainty of channel selections that appear to both CR users and the attacker. Thus, it performs poorly for control channel establishment while effectively combating jamming attacks. For CR users, PHC performs better than WoLF-PHC with low to medium PU activities due to its greedy approach. However, WoLF-PHC outperforms PHC under high PU activities due to the variable learning rates. The proposed JRCC achieves the highest ψ values among all methods by combining the advantages of PHC and WoLF-PHC. With low and medium PU activities, JRCC further outperforms PHC with cooperative gain due to action strategy coordination among CR users when valid control links are constantly available. For high PU activities where valid control channels are less likely to exist, JRCC's variable learning rates take into effect in increasing jamming resilience. For the attacker, all three MARL methods exhibit increasing jamming strength ζ under high PU activities because only a few channels are available in this case, which gives the attacker an edge to jamming. PHC generally achieves the best jamming strength ζ under low and medium

PU activities due to greediness. Jamming strength of the attacker in JRCC is smaller because it is compromised by jamming resilience of CR users and no cooperation can be performed by the attacker. For extremely high PU activities where valid CCCs are very limited for jamming, jamming strength of the attacker also decreases and approaches zero as PU becomes mostly active on all channels. This scenario shows that JRCC effectively adapts to PU activities by combining action strategy coordination and variable learning rates to enhance jamming resilience of control links for defending jamming attacks.

6.6.4 Best-Effort Cooperative Sensing

Besides the effects of PU activities, spectrum sensing errors such as false alarms and miss detection can have a large impact on JRCC performance. We first show the impact of spectrum sensing errors on jamming resilience of CR users and jamming strength of the attacker, and then show how distributed cooperative sensing with sensing decision fusion rules can mitigate the effects of false alarm and miss detection errors, and improve jamming resilience.

6.6.4.1 Impact of Spectrum Sensing Errors

Figure 6.5 shows the effects of spectrum sensing errors ($P_f = 0.1$ and/or $P_m = 0.1$ on jamming resilience/strength (ψ/ζ) for different degrees of PU activities. Note that inverted jamming strength ($\phi = 1 - \zeta$) is shown in the figure for clarity. For CR users, both false alarms and miss detection cause jamming resilience degradation under all levels of PU activities. Specifically, for low to medium PU activities where PUs are less likely to occupy the channels, miss detection results in relatively minor degradation while false alarms result in significant reduction in performance. When both types of errors are considered, the degradation is clearly dominated by false alarm errors. For high PU activities where PUs occupy channels more frequently, miss detection degrades performance slightly more than false alarms. When both

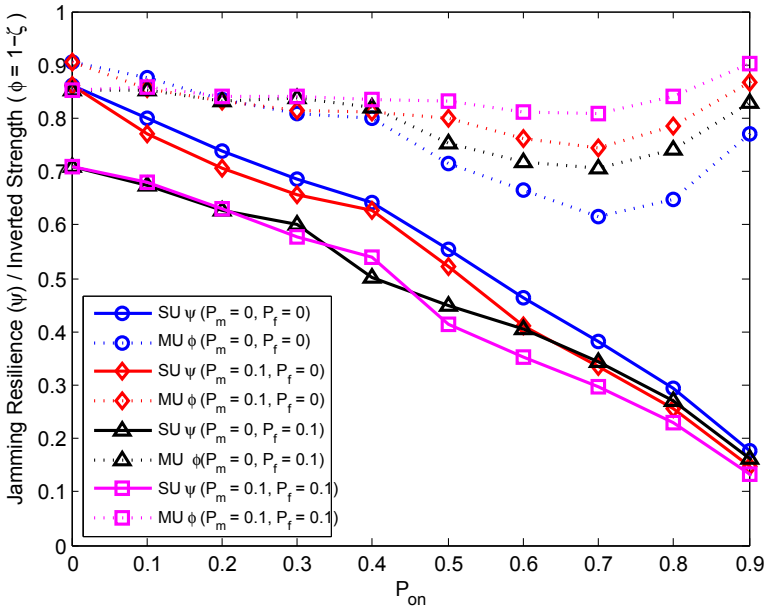


Figure 6.5: Effects of Spectrum Sensing Errors ($P_f = 0.1$ and/or $P_m = 0.1$) on Jamming Resilience and Jamming Strength for Different Values of P_{on}.

are considered, the degradations from each type of errors are accumulated. Thus, we observe that false alarms cause more damage to jamming resilience than miss detection. For the attacker, jamming strength is also compromised by both types of sensing errors under medium to high PU activities. Unlike CR users, the attacker's jamming strength suffers more from miss detection than from false alarms.

6.6.4.2 False Alarm Errors

Figure 6.6 shows jamming resilience/strength under different degrees of false alarm errors when CR users employ cooperative sensing with AND, OR, and Majority (MAJ) decision data fusion rules [3]. We include perfect and imperfect scenarios for comparison. The perfect scenarios refer to the cases where all control links are available for sharing sensing decisions among CR users while the imperfect ones are scenarios where only valid control channels from the previous stage that are not occupied by PUs in the current stage will be available for reporting sensing results. We first see that, without using cooperative sensing, jamming resilience decreases by 47% as

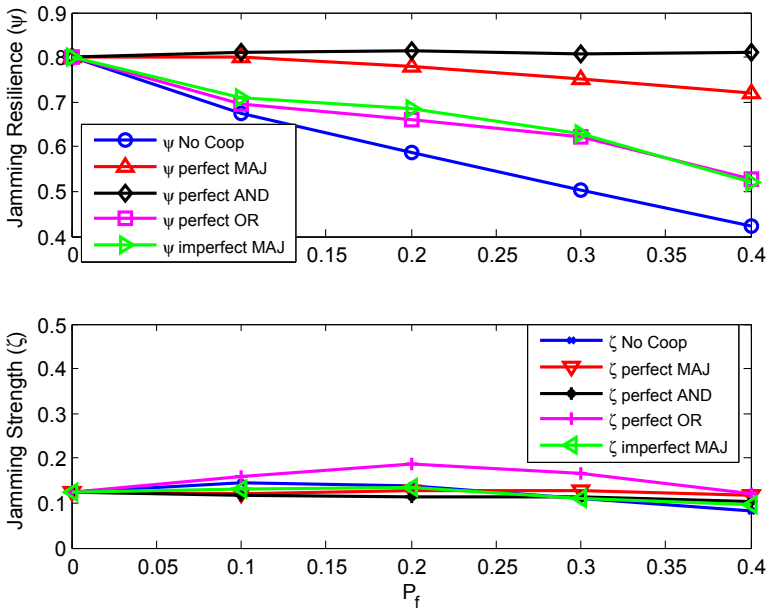


Figure 6.6: Jamming Resilience and Jamming Strength under the Impact of False Alarms ($P_{\text{on}} = 0.1$, $P_m = 0$).

P_f increases from 0 to 0.4. With cooperative sensing, performance degradation can be effectively mitigated by OR rule, significantly improved by MAJ rule, and completely recovered by AND rule in perfect CCC scenarios. This shows the importance of control channels for cooperative sensing whose performance improvement further enhances jamming resilience of control links. Since the perfect scenarios generally do not exist in hostile environment, we show that using MAJ rule can achieve similar or slightly better performance in the imperfect scenarios compared to the OR rule in the perfect scenarios. Therefore, distributed cooperative sensing can effectively mitigate the impact of false alarms to improve jamming resilience of CR users. For the attacker, the jamming strength is slightly affected by false alarms in this low PU activity scenario.

6.6.4.3 Miss Detection Errors

Figure 6.7 shows jamming resilience/strength under different degrees of miss detection errors when CR users employ cooperative sensing with AND, OR, and MAJ decision

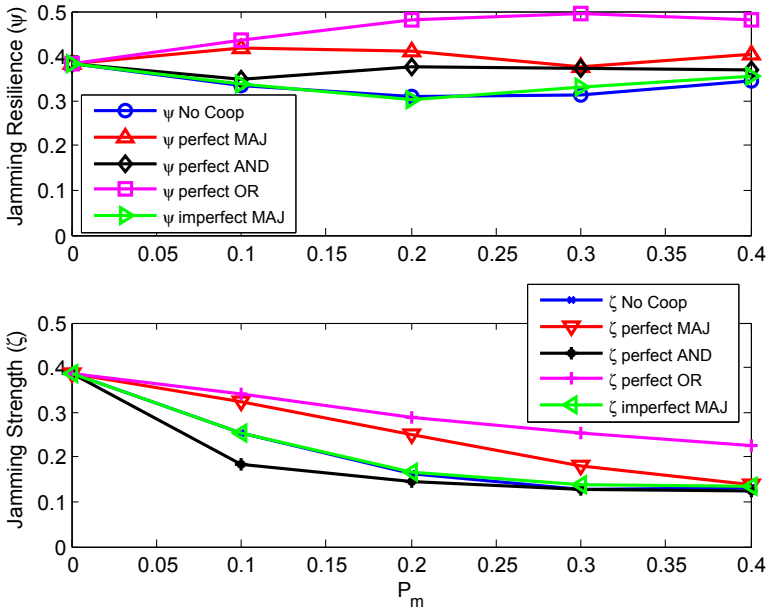


Figure 6.7: Jamming Resilience and Jamming Strength under the Impact of Miss Detection ($P_{\text{on}} = 0.7$, $P_f = 0$).

data fusion rules. Evidently, miss detection has more impact on jamming strength of the attacker than jamming resilience of CR users. The attacker's jamming strength is significantly reduced up to 65% as P_m increases from 0 to 0.4. Thus, the impact of miss detection on jamming strength of the attacker is more severe than that of false alarms. For CR users, all decision rules for cooperative sensing improve jamming resilience with OR rule the most improved, AND rule the least improved, and MAJ rule in between when control channels are perfectly available. Interestingly, using OR rules can further increase ψ as P_m increases. This is due to perfect cooperative detection by CR users and miss detection by the attacker that decreases ζ. We also show that using MAJ rule in the imperfect scenarios can achieve similar or slightly better performance compared to no cooperation case. Thus, combined with the results in Figure 6.6, we obtain that MAJ rule achieves the best tradeoff in mitigating false alarm and miss detection errors to improve jamming resilience in both perfect and imperfect control link scenarios.

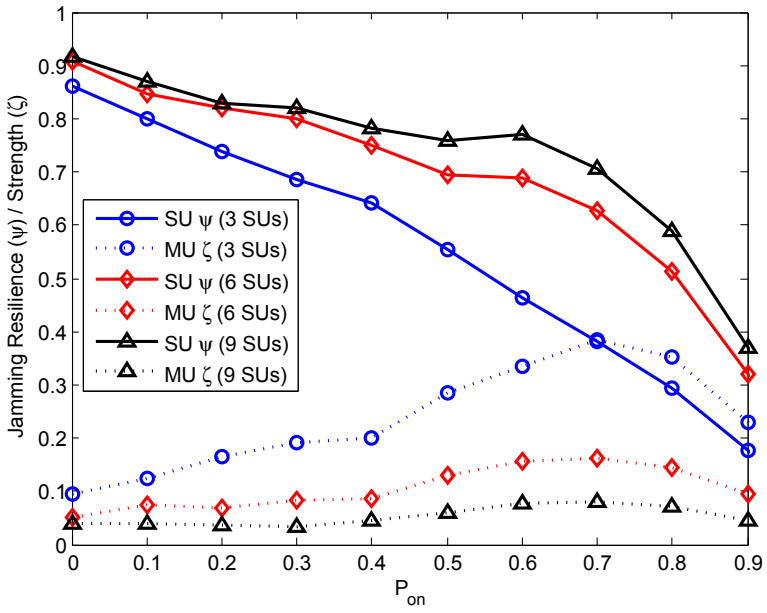


Figure 6.8: Effects of Deployment Density and Scalability on Jamming Resilience and Jamming Strength under Different Degrees of PU Activities.

6.6.5 Deployment Density and Scalability

In addition to action strategy coordination and distributed cooperative sensing, the deployment density of CR users in the jamming region is an effective countermeasure to defend jamming attacks. Figure 6.8 shows jamming resilience/strength under all levels of PU activities with different CR user deployments ($K = 3, 6, 9$) in the jamming region. We can see that jamming resilience (strength) increases (decreases) as the number of deployed CR users increases. The gain from higher deployment density is more evident under medium to high PU activities than low PU activities. Note that the increase in ψ is diminishing as the deployment density increases because it is getting more difficult for CR users to coordinate with all the neighbors with dynamic channel selection strategies and limited transmission capability. This shows the scalability of JRCC scheme in the jamming region and the effectiveness of using deployment density as a countermeasure to enhance jamming resilience of CR users and counteract jamming strength of the attacker.

CHAPTER VII

CONCLUSIONS

Cognitive radio networks have been recognized as a promising paradigm to address the spectrum under-utilization problem. To improve spectrum efficiency, many operations such as sharing data in cooperative spectrum sensing, broadcasting spectrum-aware routing information, and coordinating spectrum access rely on control message exchange on a common control channel. In this research, we propose a framework for common control channel design and analysis to address common control channel reliability issues and achieve seamless communication and spectral efficiency in CR ad hoc networks. Specifically, we address common control channel reliability issues by achieving the following three main objectives: (i) responsiveness to PU activities, (ii) robustness to channel impairments, and (iii) resilience to jamming attacks.

In Chapter 2, we first introduce the origins, the problem, the definition, and the classification of CCCs. We then discuss four major CCC design methods: sequence-based, group-based, dedicated, and underlay CCC design, their advantages and disadvantages, and existing solutions. We also discuss control channel security issues with the focus on control channel jamming. This chapter provides the fundamental knowledge of CCC design and challenges in CR ad hoc networks. In Chapter 3, we focus on the importance of CCCs in cooperation and examine the roles that CCCs play in cooperative spectrum sensing. We start with introducing cooperative spectrum sensing, its elements and framework, and achievable cooperative gain and incurred cooperation overhead. we are particularly interested in tackling CCC-related cooperation overhead issues such as reporting delay, channel impairments, and security.

We then discuss cooperative sensing security issues with the emphasis on data falsification attacks. These two chapters lay the foundation for more advanced topics and in-depth discussions about our proposed methods in later chapters.

In Chapter 4, we propose an efficient recovery control channel (ERCC) method to address the challenge of achieving responsiveness to PU activities and extending control channel coverage for reducing control efforts. By constantly updating and periodically exchanging common channel lists among neighboring nodes, our proposed method is capable of efficiently responding to primary user activity change with dynamic control channel allocation. It also balances the tradeoff between extending the coverage of common control channels for reducing control signaling efforts and selecting channels of best quality for minimizing the interference with primary users. With common channel lists constructed in each ad hoc node, the work can be extended to consider the adaptation of control channel bandwidth for a variety of control traffic loads. Furthermore, the optimal utilization of the common channel lists for both control and data channel allocations can be developed for throughput and quality-of-service analysis. Finally, cooperative spectrum sensing schemes can be incorporated into ERCC for performance improvement in realistic channel conditions.

In Chapter 5, we propose a novel cooperative sensing scheme based on reinforcement learning to improve the cooperative gain and mitigate the cooperation overhead in correlated shadowing and dynamic environment. We show that the reinforcement learning-based cooperative sensing (RLCS) method is capable of converging to an optimal solution asymptotically and enhancing rewards by using optimal stopping. The optimal solution achieved by an optimal user selection policy includes finding the optimal set of cooperating neighbors with minimum control traffic, reducing the overall reporting delay, selecting independent users for cooperation under correlated shadowing, and excluding unreliable users and data from cooperation. The results show that RLCS improves or maintains the comparable detection performance while

adapting to environmental change, such as the change in PU traffic pattern, user location, user reliability, and fading control channel condition, that may compromise the cooperative gain in cooperative sensing. This work can be extended to consider the cooperative wideband sensing scenario where multiple CR users act as fusion centers in different frequency bands to jointly improve cooperative gain and mitigate cooperation overhead with multiagent reinforcement learning.

In Chapter 6, we tackle the control channel jamming problem in CR ad hoc networks by proposing a jamming-resilient control channel (JRCC) scheme to enhance the jamming resilience of CR users and counteract the jamming strength of intelligent attackers. We model the interactions among CR users and the attacker under the impact of PU activities as a stochastic general-sum game, and analyze the gradient ascent dynamics and the convergence of the game. We also propose the JRCC algorithm to adapt to PU activities with variable learning rates and defend jamming attacks by using action strategy coordination, distributed cooperative sensing, and higher deployment density to facilitate control channel establishment, mitigate the impact of spectrum sensing errors, and counteract jamming strength of the attacker. The results demonstrate that JRCC scheme effectively combats jamming under the impact of primary user activities and spectrum sensing errors by utilizing action strategy coordination, best-effort cooperative sensing, and higher deployment density as intrusion defense countermeasures to intelligent jamming attacks. This work can be extended to the large-scale jamming scenario where control channels in CR ad hoc networks are attacked by multiple attackers whose cumulative effects on the jamming resilience of CR users in overlapping jamming regions are of great interest.

APPENDIX A

TABLES OF NOTATIONS

In this appendix, we tabulate the notations frequently used in Chapter 4, 5, and 6. Specifically, Table A.1 and A.2 list the notations used in Chapter 4, Table A.3 and A.4 list the notations used in Chapter 5, and Table A.5 and A.6 list the notations used in Chapter 6.

Table A.1: Table of Notations (A-M) for Chapter 4.

Notation	Description
A	PU and CR user deployment area
a	Decaying coefficient in the exponential correlation model
α	Death rate of PU activity (ON to OFF)
B	Channel bandwidth (control or data)
β	Birth rate of PU activity (OFF to ON)
C	CCC allocation state
C_i/C_j	The i^{th}/j^{th} licensed channel
Ch_k	The CCC allocated to neighbor k
δ_{DL}	Threshold for detecting unreliable CR users
d_{ij}	Distance between CR user i and j
d_{pi}	Distance between the PU and CR user i
D_p	Density of PU deployment
D_s	Density of CR user deployment
γ_i	Accumulated PU interference on channel C_i
γ_{pu}	Sensing threshold for PU receive power
γ_{su}	Sensing threshold for CR user receive power
L_{C_i}	Common channel list of CR user i
L_{NB}	List of neighbors with allocated CCCs
L_P	Preferred channel list from local sensing
L_{CC}	The intersection of two broadcast CCLs L_{C_i} and L_{C_k}
L_R	List of channels for control radio channel hopping

Table A.2: Table of Notations (N-Z) for Chapter 4.

Notation	Description
N_{best}	Number of CCC links with the best allocated CCC
N_c	Number of licensed channels
N_{disc}	Number of established CCC links
N_r	Number of control capacity regions
N_k	Number of neighbors
N_m	Maximum number of channel switches during recovery
N_p	Number of PUs
N_s	Number of CR users
N_{tot}	Number of total available CCC links
p	Probability of selecting the common channel C_i
P_I	Interference power observed at a PU
P_{su}	CR user transmit power
r_k	CCL broadcast rate of CR user k
R	CCC recovery state
R_c	Sum-rate capacity of all control channels
$R_j^k(q)$	Maximum control throughput on C_j at CR user k in region q
R_p	PU transmission range
R_s	CR user transmission range
r_s	Periodic spectrum sensing frequency
S_m	The m^{th} channel in the channel hopping sequence
σ_{dB}	Log-normal shadow fading dB-spread
t_{disc}	Maximum duration for initial neighbor discovery
t_p	PU ON/OFF period
t_C/T_C	CCC allocation time
t_R/T_R	CCC recovery time
w_{ij}	The weight for the CCC link between CR user i and j
W_i	The weight of channel C_i for channel ordering

Table A.3: Table of Notations (A-M) for Chapter 5.

Notation	Description
a	Decaying coefficient of exponential correlation model
$\mathcal{A}_j$	Set of actions available in state j
A_j	The j-th boundary of SNR regions
a_k	Action selected in s_k
α	Learning rate
$\mathcal{C}$	Set of cooperating CR users
$C_{d_{k+1}}$	Delay cost
$C_{\rho_{k+1}}$	Correlation cost
$D(p \parallel q)$	Kullback-Leibler distance between p and q
D_c	De-correlation distance
d_{ij}	Distance between CR user i and j
$d_{k,i}$	Reporting delay of CR user i in stage k
$\mathcal{D}_k$	Set of selected CR users from s_0 to s_k
γ	Discount factor
γ_b	Received SNR
γ_{pl}	Path loss exponent
h_k	History of state-action sequence up to s_k
k	Stage or time index
K	Number of selected neighbors for cooperation
$\hat{K}$	Rate of convergence
L	Number of cooperating neighbors
λ_0	Cooperative decision threshold
λ_{th}	Local decision threshold
μ_k	Decision rule mapping h_k to a_k

Table A.4: Table of Notations (N-Z) for Chapter 5.

Notation	Description
N	Number of episodes
$\mathcal{P}^*$	Optimal set of selected CR users
P_d	Local probability of detection
P_e	Error probability of control channels
P_f	Local probability of false alarm
π^*	Optimal policy or strategy
$Q(s,a)$	State-action value for choosing action a in state s
Q_f	Probability of false alarm for cooperative decisions
Q_m	Probability of miss detection for cooperative decisions
R^n	Expected cumulative reward of episode n
r_{k+1}	Immediate reward
ρ_{ij}	Correlation coefficient between CR user i and j
r_b	Birth rate (OFF to ON) of PU activity
r_d	Death rate (ON to OFF) of PU activity
$r_{d_{k+1}}$	Reward attributed to delay cost
$r_{\rho_{k+1}}$	Reward attributed to correlation cost
σ_1	Lognormal dB-spread
s_k	State of cooperative sensing decision process in stage k
T^*	Optimal stopping time
T_{lim}	Total reporting delay constraint
τ^n	Temperature of softmax action strategy in episode n
t_{d_i}	Reporting delay of CR user i
$\mathcal{U}$	Set of selected uncorrelated CR users
u_i	Local decision of CR user i
u_0	Cooperative decision
Y	Cumulative reward
y_i	Observation of CR user i

Table A.5: Table of Notations (A-M) for Chapter 6.

Notation	Description
$\mathfrak{A}$	Area of jamming region
$\mathcal{A}$	Action space of JRCC game
$\mathcal{A}_n(s)$	Set of actions available to player n in state s
a_n^t	Action of player n in stage t
α	Step size
B_k	Number of neighbors of CR user k
$C_{k,j}$	Number of valid CCCs between CR user k and j
c_j	Jammed common control channel
$C_{\mathcal{K}}$	Number of establish control links in $\mathfrak{A}$
C_p	Set of PU-free channels
c_s	Selected common control channel
c_v	Valid common control channel
D	Distance between CR user and the attacker in $\mathfrak{A}$
$\mathbf{\Delta}$	Learning rate matrix
δ_i	Learning rate of CR user i
$\mathcal{D}_p$	Set of PU-occupied channels
$\Delta(x)$	Probability distribution of x
$D_{j,i}$	Number of jammed CCCs in i-th control link
$D_{\mathcal{K}}$	Number of jammed control links in $\mathfrak{A}$
D_K	CR user deployment density in $\mathfrak{A}$
d_k	Distance from a CR user to its k-th nearest neighbor
ϵ	Small probability for epsilon-greedy action selection
η	Path loss exponent
Γ	JRCC game
γ	Discount factor of JRCC algorithm
γ_k	SINR at CR user k
γ_{th}	SINR threshold for decoding messages
h_{ik}	Channel gain between player i and k
H_1/H_0	Alternative/null hypothesis
J	Number of attackers
J_c	Number of jammed CCCs
$\mathcal{K}$	Set of CR users or the entire CR ad hoc network
K	Number of CR users in JRCC game
$L_{\mathcal{K}}$	Number of possible control links in $\mathfrak{A}$
M	Number of actions available in state s, $\lvert\mathcal{A}_n(s)\rvert$

Table A.6: Table of Notations (N-Z) for Chapter 6.

Notation	Description
N	Number of players in JRCC game
N_p	Number of licensed channels
N_s	Maximum number of selected channels for transmission
n_s	Number of eigenvalues associated with stable eigenspaces
N_i	Number of channels selected by player i for transmission
n_u	Number of eigenvalues associated with unstable eigenspaces
J_c	Number of jammed CCCs
N_c	Number of selected CCCs
$\mathbf{P}(\mathbf{x})$	First recurrence map or Poincaré map of point $\mathbf{x}$
P_c	Number of PU-occupied selected CCCs
P_f	Probability of false alarm
P_i	Transmission power of player i
P_m	Probability of miss detection
P_{off}	Probability of PU OFF state
P_{on}	Probability of PU ON state
p_ζ	Value of P_{on} where jamming strength reaches the maximum
ϕ	Inverted jamming strength
π_n	Action selection strategy of player n
$\pi_{n,m}^t$	Probability of player n selecting the m-th action in stage t
ψ	Jamming resilience of CR users
Q_f	Probability of false alarm for cooperative decisions
Q_m	Probability of miss detection for cooperative decisions
R^{π_n}	Average reward of player n achieved by strategy π_n
R_j	Radius of jamming region $\mathfrak{A}$
R_n^t	Expected reward in stage t
r_b	Birth rate (OFF to ON) of PU activities
r_d	Death rate (ON to OFF) of PU activities
r_n	Immediate reward of player n
$\mathcal{S}$	State space of JRCC game
s	State of JRCC game
σ^2	Noise power
U_c	Number of valid CCCs
V^s	Stable eigenspaces
V^u	Unstable eigenspaces
$\mathbf{x}_n$	Strategy matrix of player n
xi	Similarity of two strategies
ζ	Jamming strength of attackers

APPENDIX B

TEMPORAL-DIFFERENCE LEARNING

The temporal-differnece (TD) methods in reinforcement learning facilitate the evaluation of the quality of states and the tradeoff between exploration and exploitation. In Chapter 5, we utilize three TD methods: *Q-learning*, *Sarsa*, and *Actor-Critic* methods [85] for evaluating the performance of RLCS. These algorithms are summarized as follows:

B.1 Q-Learning

Q-learning is an off-policy TD method that utilizes separate policies for evaluating the quality of states and for making decisions. In each state s_k, the agent chooses an action a_k (the next selected cooperating CR user) based on an action selection strategy, such as ϵ-greedy and softmax approach, observes the reward r_{k+1} and the next state s_{k+1}, and updates the state-action value function, called *Q-factors*, for current state s_k as follows:

$$Q(s_k, a_k) \leftarrow (1-\alpha)Q(s_k, a_k) + \alpha\big[r_{k+1} + \gamma \max_{a_{k+1}} Q(s_{k+1}, a_{k+1})\big] \tag{B.1}$$

where $\alpha \in (0, 1)$ and $\gamma \in (0, 1]$ are the learning rate and the discount factor, respectively. The state update, $s_k \leftarrow s_{k+1}$, follows. These steps of action selection, observation, and $Q(s, a)$ and state updates are repeated all over again in each state. The state-action value for the terminal state is zero.

B.2 Sarsa

Sarsa is an on-policy TD method that utilizes the state-action pair for transitions. Hence, the policy to be evaluated and improved is also used in making decisions.

Similar to Q-learning, the agent initially chooses an action and observes the reward and the next state. Unlike Q-learning, the agent takes the next-state action selected in the previous state as the current action a_k in state s_k and observes the reward r_{k+1} and the next state s_{k+1}. The agent selects next state action a_{k+1} for state s_{k+1} by using the softmax approach in state s_k and updates the state-action value function for the current state s_k as follows:

$$Q(s_k, a_k) \leftarrow (1-\alpha)Q(s_k, a_k) + \alpha\big[r_{k+1} + \gamma Q(s_{k+1}, a_{k+1})\big] \tag{B.2}$$

where α and γ are the learning rate and the discount factor, respectively. The update of the state-action pair follows: $s_k \leftarrow s_{k+1}$ and $a_k \leftarrow a_{k+1}$. These steps of taking actions, observation, next-state action selection, and $Q(s, a)$ and state-action pair updates are repeated in each state.

B.3 Actor-Critic

In the actor-critic method, the agent consists of an *actor* that chooses actions and a *critic* that evaluates those actions and the value of states. In each state, the actor chooses an action and observes the reward r_{k+1} and next state s_{k+1}. The critic estimates a temporal-difference (TD) error, δ_k, updates the value function, and sends the TD error to *criticize* the actor's action preference. Specifically, the critic evaluates the TD error, δ_k, by using the immediate reward, r_{k+1}, and the next-state value function estimate, $V(s_{k+1})$, in state s_k as follows:

$$\delta_k = r_{k+1} + \gamma V(s_{k+1}) - V(s_k) \tag{B.3}$$

where γ is the discount factor. On the other hand, the actor chooses an action based on the Softmax approach as follows:

$$\pi(s_k, x_i) = \frac{e^{p(s_k, x_i)}}{\sum_{j=1}^{N_a} e^{p(s_k, x_j)}}, \qquad i = 1, \ldots, N_a \tag{B.4}$$

where $\pi(s_k, a_k)$ indicates the preference for action $a_k = x_i$ in state s_k. Both the critic and the actor use the TD error for update. The critic updates the value function:

$V(s_k) \leftarrow V(s_k) + \alpha\delta_k$ in which α is the learning rate. Similarly, the actor updates its preference as follows: $p(s_k, a_k) \leftarrow p(s_k, a_k) + \alpha\delta_k$.

APPENDIX C

STOCHASTIC GAME

The stochastic game $\mathcal{G}$ with N players is the multi-player generalization of a single-player Markov decision process (MDP) [12], which is a tuple: $\langle \mathcal{S}, \mathcal{A}, f_p, f_R \rangle$ where $\mathcal{S}$ is the set of states, $\mathcal{A} = \{\mathcal{A}(1), \ldots, \mathcal{A}(|\mathcal{S}|)\}$ is the set of action spaces in which $A(s)$, $s \in \mathcal{S}$ is the joint action space of all players' action spaces in state s given by $A(s) = \mathcal{A}_1(s) \times \cdots \times \mathcal{A}_N(s)$, $f_p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state transition probability function, and $f_R = f_{R_1}, \ldots, f_{R_N}$ is the collection of reward functions of all N players in which $f_{R_n} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function of player n. The game is repeatedly played for infinite number of stages or for finite number of stages until the attackers are removed.

At each stage of $\mathcal{G}$, each player n observes the state changes based on state transition probabilities, selects an action according to its action selection strategy $\pi_n(s)$, and receives a reward (also called payoff) $R_n(s, \mathcal{A}(s))$ as the result of the state changes and the joint actions from all players. Although each player may not be able to observe all the states and all the actions of other players, the objective of the game is for each player n to find the optimal strategy π_i^* to maximize its expected future discounted reward. Thus, the optimal policy for player n is given by

$$\pi_n^*(s) = \max_{\pi_i} \lim_{T \to \infty} \left[\frac{1}{T} \sum_{t=1}^{T} E_{\pi_n} \left[R_t^{\pi_n} \right] \right] \tag{C.1}$$

where $R_t^{\pi_n}$ is the player n's reward received in stage t.

REFERENCES

[1] Akyildiz, I. F., Lee, W.-Y., and Chowdhury, K. R., "CRAHNs: Cognitive radio ad hoc networks," *Ad Hoc Networks*, vol. 7, pp. 810–836, July 2009.

[2] Akyildiz, I. F., Lee, W.-Y., Vuran, M. C., and Mohanty, S., "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, pp. 2127–2159, September 2006.

[3] Akyildiz, I. F., Lo, B. F., and Balakrishnan, R., "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical Communication*, vol. 4, pp. 40–62, March 2011.

[4] Arrowsmith, D. K. and Place, C. M., *Dynamical Systems*. London, UK: Chapman & Hall, 1992.

[5] Asterjadhi, A., Baldo, N., and Zorzi, M., "A distributed network coded control channel for multihop cognitive radio networks," *IEEE Network*, vol. 23, pp. 26–32, July-August 2009.

[6] Baldo, N., Asterjadhi, A., and Zorzi, M., "Dynamic spectrum access using a network coded cognitive control channel," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 2575–2587, August 2010.

[7] Berthold, U., Fu, F., van der Schaar, M., and Jondral, F. K., "Detection of spectral resources in cognitive radios using reinforcement learning," in *Proc. of IEEE Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–5, October 2008.

[8] Bian, K., Park, J.-M., and Chen, R., "A quorum-based framework for establishing control channels in dynamic spectrum access networks," in *Proc. of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 25–36, September 2009.

[9] Bian, K., Park, J.-M., and Chen, R., "Control channel establishment in cognitive radio networks using channel hopping," *IEEE Journal on Selected Areas in Communications*, vol. 29, pp. 689–703, April 2011.

[10] Bowling, M. and Veloso, M., "Multiagent learning using a variable learning rate," *Artificial Intelligence*, vol. 136, pp. 215–250, April 2002.

[11] Breiman, L., *Probability*. Boston, MA: Addison-Wesley, 1968.

[12] Busoniu, L., Babuska, R., and De Schutter, B., "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, pp. 156–172, March 2008.

[13] Cabric, D., Mishra, S. M., and Brodersen, R. W., "Implementation issues in spectrum sensing for cognitive radios," in *Proc. of 38th Asilomar Conference on Signals, Systems, and Computers*, pp. 772–776, November 2004.

[14] Cabric, D., Mishra, S. M., Willkomm, D., Brodersen, R., and Wolisz, A., "A cognitive radio approach for usage of virtual unlicensded spectrum," in *Proc. of 14th IST Mobile Wireless Communications Summit*, June 2005.

[15] Chan, A., Liu, X., Noubir, G., and Thapa, B., "Broadcast control channel jamming: Resilience and identification of traitors," in *Proc. of IEEE International Symposium on Information Theory (ISIT)*, pp. 2496–2500, June 2007.

[16] Chen, R., Park, J.-M., and Bian, K., "Robust distributed spectrum sensing in cognitive radio networks," in *Proc. of IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1876 –1884, April 2008.

[17] Chen, R., Park, J.-M., and Reed, J., "Defense against primary user emulation attacks in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 25–37, January 2008.

[18] Chen, T., Zhang, H., Katz, M. D., and Zhou, Z., "Swarm intelligence based dynamic control channel in CogMesh," in *Proc. of IEEE International Conference on Communications (ICC)*, pp. 123–128, May 2008.

[19] Chen, T., Zhang, H., Maggio, G. M., and Chlamtac, I., "CogMesh: a cluster-based cognitive radio network," in *Proc. of IEEE Dynamic Spectrum Access Networks (DySPAN)*, pp. 168–178, April 2007.

[20] Chen, T., Zhang, H., Maggio, G. M., and Chlamtac, I., "Topology management in CogMesh a cluster-based cognitive radio mesh network," in *Proc. of IEEE International Conference on Communications (ICC)*, pp. 6516–6521, June 2007.

[21] Chowdhury, K. and Akyldiz, I., "OFDM-based common control channel design for cognitive radio ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 228–238, February 2011.

[22] Cormio, C. and Chowdhury, K. R., "Common control channel design for cognitive radio wireless ad hoc networks using adaptive frequency hopping," *Ad Hoc Networks*, vol. 8, pp. 430–438, June 2010.

[23] DaSilva, L. A. and Guerreiro, I., "Sequence-based rendezvous for dynamic spectrum access," in *Proc. of IEEE Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–7, October 2008.

[24] Di Felice, M., Chowdhury, K. R., Meleis, W., and Bononi, L., "To sense or to transmit: A learning-based spectrum management scheme for cognitive radiomesh networks," in *Proc. of IEEE Workshop on Wireless Mesh Networks (WIMESH)*, pp. 1–6, June 2010.

[25] Di Renzo, M., Graziosi, F., and Santucci, F., "Cooperative spectrum sensing in cognitive radio networks over correlated log-normal shadowing," in *Proc. of IEEE Vehicular Technology Conference (VTC2009-Spring)*, January 2009.

[26] Di Renzo, M., Imbriglio, L., Graziosi, F., and Santucci, F., "Cooperative spectrum sensing over correlated log-normal sensing and reporting channels," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, pp. 1–8, November 2009.

[27] ECMA, "MAC and PHY for operation in TV white space," *ECMA-392*, June 2012.

[28] FCC, "Notice of proposed rule making and order," *ET Docket No. 03-222*, December 2003.

[29] FCC, "Unlicensed operation in the tv broadcast bands, second memorandum opinion and order," *FCC 10-174*, September 2010.

[30] FCC, "Unlicensed operation in the tv broadcast bands, third memorandum opinion and order," *FCC 12-36*, April 2012.

[31] Frey, B. J. and Dueck, D., "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, February 2007.

[32] Ghasemi, A. and Sousa, E. S., "Collaborative spectrum sensing for opportunistic access in fading environments," in *Proc. of IEEE Dynamic Spectrum Access Networks (DySPAN)*, pp. 131–136, November 2005.

[33] Ghosh, C., Roy, S., and Cavalcanti, D., "Coexistence challenges for heterogeneous cognitive wireless networks in TV white spaces," *IEEE Wireless Communications*, vol. 18, pp. 22–31, August 2011.

[34] Goldsmith, A., *Wireless Communications*. New York, NY: Cambridge University Press, 2005.

[35] Gudmundson, M., "Correlation model for shadow fading in mobile radio systems," *Electronics Letters*, vol. 27, pp. 2145–2146, November 1991.

[36] Hamdaoui, B. and Shin, K. G., "OS-MAC: an efficient MAC protocol for spectrum-agile wireless networks," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 915–930, August 2008.

[37] Haykin, S., "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 201–220, February 2005.

[38] IEEE, "Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 22: Cognitive wireless ran medium access control (mac) and physical layer (phy) specifications: Policies and procedures for operation in the tv bands," *IEEE Std 802.22-2011*, pp. 1–680, 2011.

[39] Jia, J., Zhang, Q., and Shen, X., "HC-MAC: a hardware constrained cognitive MAC for efficient spectrum management," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 106–117, January 2008.

[40] Jing, X. and Raychaudhuri, D., "Spectrum co-existence of IEEE 802.11b and 802.16a networks using the CSCC etiquette protocol," in *Proc. of IEEE Dynamic Spectrum Access Networks (DySPAN)*, pp. 243–250, November 2005.

[41] Kakumanu, S. and Sivakumar, R., "Glia: a practical solution for effective high datarate wifi-arrays," in *Proc. of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 229–240, September 2009.

[42] Kaligineedi, P., Khabbazian, M., and Bhargava, V., "Secure cooperative sensing techniques for cognitive radio systems," in *Proc. of IEEE International Conference on Communications (ICC)*, pp. 3406–3410, May 2008.

[43] Kaligineedi, P., Khabbazian, M., and Bhargava, V., "Malicious user detection in a cognitive radio cooperative sensing system," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 2488–2497, August 2010.

[44] Kondareddy, Y. R. and Agrawal, P., "Synchronized MAC protocol for multi-hop cognitive radio networks," in *Proc. of IEEE International Conference on Communications (ICC)*, pp. 3198–3202, May 2008.

[45] Kondareddy, Y., Agrawal, P., and Sivalingam, K., "Cognitive radio network setup without a common control channel," in *Proc. of IEEE Military Communications Conference (MILCOM)*, pp. 1–6, November 2008.

[46] Lazos, L., Liu, S., and Krunz, M., "Spectrum opportunity-based control channel assignment in cognitive radio networks," in *Proc. of IEEE Sensor, Mesh, and Ad Hoc Communications and Networks (SECON)*, pp. 1–9, June 2009.

[47] Lazos, L., Liu, S., and Krunz, M., "Mitigating control-channel jamming attacks in multi-channel ad hoc networks," in *Proc. of the Second ACM Conference on Wireless network security (WiSec)*, pp. 169–180, March 2009.

[48] Le, L. and Hossain, E., "OSA-MAC: A MAC protocol for opportunistic spectrum access in cognitive radio networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1426–1430, April 2008.

[49] Lee, W.-Y. and Akyildiz, I. F., "Optimal spectrum sensing framework for cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 7, pp. 3845–3857, October 2008.

[50] Li, H. and Han, Z., "Dogfight in spectrum: Jamming and anti-jamming in multichannel cognitive radio systems," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, December 2009.

[51] Li, M., Koutsopoulos, I., and Poovendran, R., "Optimal jamming attack strategies and network defense policies in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 1119–1133, August 2010.

[52] Liu, S., Lazos, L., and Krunz, M., "Cluster-based control channel allocation in opportunistic cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 11, pp. 1436–1449, October 2012.

[53] Lo, B. F. and Akyildiz, I. F., "Reinforcement learning-based cooperative sensing in cognitive radio ad hoc networks," in *Proc. of IEEE Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 2244–2249, September 2010.

[54] Lo, B. F., Akyildiz, I. F., and Al-Dhelaan, A. M., "Efficient recovery control channel design in cognitive radio ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 4513–4526, November 2010.

[55] Lo, B. F. and Akyildiz, I., "Multiagent jamming-resilient control channel game for cognitive radio ad hoc networks," in *Proc. of IEEE International Conference on Communications (ICC)*, pp. 1821–1826, June 2012.

[56] Lo, B. F. and Akyildiz, I., "Jamming-resilient control channels for intrusion defense in cognitive radio ad hoc networks," *submitted for publication*, November 2013.

[57] Lo, B. F., "A survey on common control channel design for cognitive radio networks," *Physical Communication*, vol. 4, pp. 26–39, March 2011.

[58] Lo, B. F. and Akyildiz, I. F., "Reinforcement learning for cooperative sensing gain in cognitive radio ad hoc networks," *Wireless Networks*, vol. 19, pp. 1237–1250, August 2013.

[59] Ma, J., Zhao, G., and Li, Y., "Soft combination and detection for cooperative spectrum sensing in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 7, pp. 4502–4507, November 2008.

[60] Ma, L., Shen, C.-C., and Ryu, B., "Single-radio adaptive channel algorithm for spectrum agile wireless ad hoc networks," in *Proc. of IEEE Dynamic Spectrum Access Networks (DySPAN)*, pp. 547–558, April 2007.

[61] Masri, A., Chiasserini, C.-F., Casetti, C., and Perotti, A., "Common control channel allocation in cognitive radio networks through UWB multihop communications," in *Proc. of the First Nordic Workshop on Cross-Layer Optimization in Wireless Networks*, April 2010.

[62] Masri, A., Chiasserini, C.-F., and Perotti, A., "Control information exchange through UWB in cognitive radio networks," in *Proc. of IEEE International Symposium on Wireless Pervasive Computing (ISWPC)*, pp. 110–115, May 2010.

[63] Mishra, S., Sahai, A., and Brodersen, R., "Cooperative sensing among cognitive radios," in *Proc. of IEEE International Conference on Communications (ICC)*, vol. 4, pp. 1658–1663, June 2006.

[64] Mo, J., So, H.-S., and Walrand, J., "Comparison of multichannel mac protocols," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 50–65, January 2008.

[65] Motamedi, A. and Bahai, A., "Mac protocol design for spectrum-agile wireless networks: Stochastic control approach," in *Proc. of IEEE Dynamic Spectrum Access Networks (DySPAN)*, pp. 448–451, April 2007.

[66] Oksanen, J., Lundén, J., and Koivunen, V., "Reinforcement learning-based multiband sensing policy for cognitive radios," in *Proc. of the Second International Workshop on Cognitive Information Processing (CIP)*, pp. 316–321, June 2010.

[67] Oksanen, J., Lundén, J., and Koivunen, V., "Reinforcement learning method for energy efficient cooperative multiband spectrum sensing," in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 59–64, August 2010.

[68] Omiyi, P., Haas, H., and Auer, G., "Analysis of TDD cellular interference mitigation using busy-bursts," *IEEE Transactions on Wireless Communications*, vol. 6, pp. 2721–2731, July 2007.

[69] Pawelczak, P., Pollin, S., So, H.-S. W., Motamedi, A., Bahai, A., Prasad, R. V., and Hekmat, R., "State of the art in opportunistic spectrum access medium access control design," in *Proc. of the Third International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom)*, pp. 1–6, May 2008.

[70] Pawelczak, P., Pollin, S., So, H.-S., Bahai, A., Prasad, R., and Hekmat, R., "Comparison of opportunistic spectrum multichannel medium access

control protocols," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–6, December 2008.

[71] Petracca, M., Pomposini, R., Mazzenga, F., Giuliano, R., and Vari, M., "An always available control channel for cooperative sensing in cognitive radio networks," in *Proc. of IFIP Wireless Days (WD)*, pp. 1–5, October 2010.

[72] Poor, H. V. and Hadjiliadis, O., *Quickest Detection*. Cambridge, UK: Cambridge University Press, 2009.

[73] Puterman, M. L., *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY: John Wiley & Sons, 1994.

[74] Quan, Z., Cui, S., and Sayed, A., "Optimal linear cooperation for spectrum sensing in cognitive radio networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, pp. 28–40, February 2008.

[75] Raychaudhuri, D. and Jing, X., "A spectrum etiquette protocol for efficient coordination of radio devices in unlicensed bands," in *Proc. of IEEE Personal, Indoor, and Mobile Radio Communications (PIMRC)*, vol. 1, pp. 172–176, September 2003.

[76] Safdar, G. and O'Neill, M., "Common control channel security framework for cognitive radio networks," in *Proc. of IEEE Vehicular Technology Conference (VTC2009-Spring)*, pp. 1–5, April 2009.

[77] Sahin, M. E. and Arslan, H., "System design for cognitive radio conmunications," in *Proc. of the First International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom)*, pp. 1–5, Jun. 2006.

[78] Selen, Y., Tullberg, H., and Kronander, J., "Sensor selection for cooperative spectrum sensing," in *Proc. of IEEE Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–11, October 2008.

[79] Shiryayev, A. N., *Optimal Stopping Rules*. New York, NY: Springer-Verlag, 1978.

[80] Singh, S., Kearns, M., and Mansour, Y., "Nash convergence of gradient dynamics in general-sum games," in *Proc. of the 16th Conference on Uncertainty in Artificial Intelligence*, pp. 541–548, June 2000.

[81] So, J. and Vaidya, N. H., "Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," in *Proc. of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 222–233, May 2004.

[82] Song, C. and Zhang, Q., "Sliding-window algorithm for asynchronous cooperative sensing in wireless cognitive networks," in *Proc. of IEEE International Conference on Communications (ICC)*, pp. 3432–3436, May 2008.

[83] Su, H. and Zhang, X., "Cross-layer based opportunistic mac protocols for qos provisionings over cognitive radio wireless networks," *IEEE Journal on Selected Areas in Communications,*, vol. 26, pp. 118–129, Jan. 2008.

[84] Sun, C., Zhang, W., and Letaief, K., "Cooperative spectrum sensing for cognitive radios under bandwidth constraints," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–5, March 2007.

[85] Sutton, R. and Barto, A., *Reinforcement Learning: An Introduction*. Cambridge, MA: The MIT Press, 1988.

[86] Tague, P., Li, M., and Poovendran, R., "Probabilistic mitigation of control channel jamming via random key distribution," in *Proc. of IEEE Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–5, September 2007.

[87] Tague, P., Li, M., and Poovendran, R., "Mitigation of control channel jamming under node capture attacks," *IEEE Transactions on Mobile Computing*, vol. 8, pp. 1221–1234, September 2009.

[88] Theis, N., Thomas, R., and DaSilva, L., "Rendezvous for cognitive radios," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 216–227, February 2011.

[89] Unnikrishnan, J. and Veeravalli, V. V., "Cooperative sensing for primary detection in cognitive radio," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, pp. 18–27, February 2008.

[90] Varshney, P. K., *Distributed Detection and Data Fusion*. Springer-Verlag New York, 1997.

[91] Visotsky, E., Kuffner, S., and Peterson, R., "On collaborative detection of TV transmissions in support of dynamic spectrum sharing," in *Proc. of IEEE Dynamic Spectrum Access Networks (DySPAN)*, pp. 338–345, November 2005.

[92] Visser, F. E., Janssen, G. J., and Pawelczak, P., "Multinode spectrum sensing based on energy detection for dynamic spectrum access," in *Proc. of IEEE Vehicular Technology Conference (VTC2008-Spring)*, pp. 1394–1398, May 2008.

[93] Vucevic, N., Akyildiz, I. F., and Perez-Romero, J., "Dynamic cooperator selection in cognitive radio networks," *Ad Hoc Networks*, vol. 10, pp. 789–802, July 2012.

[94] Wang, B., Wu, Y., Liu, K., and Clancy, T., "An anti-jamming stochastic game for cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, pp. 877–889, April 2011.

[95] Wang, H. S. and Moayeri, N., "Finite-state markov channel–A useful model for radio communication channels," *IEEE Transactions on Vehicular Technology*, vol. 44, pp. 163–171, February 1995.

[96] Wasden, D., Moradi, H., and Farhang-Boroujeny, B., "Design and implementation of an underlay control channel for cognitive radios," *IEEE Journal on Selected Areas in Communications*, vol. 30, pp. 1875–1889, November 2012.

[97] Watkins, C. J. C. H. and Dayan, P., "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, May 1992.

[98] Xin, C. and Cao, X., "A cognitive radio network architecture without control channel," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, December 2009.

[99] Yu, F., Tang, H., Huang, M., Li, Z., and Mason, P., "Defense against spectrum sensing data falsification attacks in mobile ad hoc networks with cognitive radios," in *Proc. of IEEE Military Communications Conference (MILCOM)*, pp. 1–7, October 2009.

[100] Zhang, W. and Letaief, K., "Cooperative spectrum sensing with transmit and relay diversity in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 7, pp. 4761–4766, December 2008.

[101] Zhao, J., Zheng, H., and Yang, G.-H., "Distributed coordination in dynamic spectrum allocation networks," in *Proc. of IEEE Dynamic Spectrum Access Networks (DySPAN)*, pp. 259–268, November 2005.

[102] Zhao, Q. and Sadler, B., "A survey of dynamic spectrum access," *IEEE Signal Processing Magazine*, vol. 24, pp. 79–89, May 2007.

[103] Zhou, X., Li, G. Y., Li, D., Wang, D., and Soong, A. C. K., "Bandwidth efficient combination for cooperative spectrum sensing in cognitive radio networks," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 3126–3129, March 2010.

[104] Zhou, X., Ma, J., Li, G., Kwon, Y., and Soong, A., "Probability-based combination for cooperative spectrum sensing," *IEEE Transactions on Communications*, vol. 58, pp. 463–466, February 2010.

[105] Zhu, J., Waltho, A., Yang, X., and Guo, X., "Multi-radio coexistence: Challenges and opportunities," in *Proc. of International Conference on Computer Communications and Networks (ICCCN)*, pp. 358–364, August 2007.

[106] Zhu, Q., Li, H., Han, Z., and Basandar, T., "A stochastic game model for jamming in multi-channel cognitive radio systems," in *Proc. of IEEE International Conference on Communications (ICC)*, pp. 1–6, May 2010.